WEIR, JOHN B., II, Ph.D. Enemy Item Detection using Data Mining Methods. (2019) Directed by Dr. Richard Luecht, 114 pp.

Enemy items are any two items that should not appear on the same test form. These items may address the same material, or one may provide clues about the answer to another. Most enemy item pairs are identified before forms are published; subject matter experts (SMEs) manually review forms for enemy pairs, a process that can be both cognitively taxing and expensive. Some have suggested statistical approaches for identifying enemy item pairs; for instance, response data might show violations of local independence caused by clueing. One drawback, however, is that these are *post hoc* tests: the forms must have been administered to a sufficient number of examinees.

This study proposed a method of identifying enemy item pairs that capitalized on two data mining approaches: latent Dirichlet allocation (LDA), an unsupervised topic model, and a random forest classifier, a supervised ensemble learning algorithm. Output from the LDA model was used to calculate the Jensen-Shannon distance (JSD) between items. Random forests were trained with and without the JSD, as well as several other item-level variables. Item pairs were scored using the resulting random forest classifiers, and SMEs evaluated the output. The random forest classifier was then retrained using input from the SMEs.

This study suggests that random forest models can be useful in the identification of enemy item pairs; information derived from the LDA topic model improves the performance of the random forest classifier, and integrating feedback from SMEs further improves the performance.

ENEMY ITEM DETECTION USING DATA MINING METHODS

by

John B. Weir II

A Dissertation Submitted to
the Faculty of The Graduate School at
The University of North Carolina at Greensboro
in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Greensboro
2019

Approved by

_____
Committee Chair

To Kathryn, Edie, and Frances, who worked harder than I did.

APPROVAL PAGE

This dissertation written by John B. Weir II has been approved by the following committee of the Faculty of The Graduate School at The University of North Carolina at Greensboro.

Committee Chair _____

Committee Members _____

_____

_____

_____

Date of Acceptance by Committee

_____

Date of Final Oral Examination

ACKNOWLEDGEMENTS

I would like to express my deepest appreciation to Dr. Ric Luecht, a patient and encouraging advisor, and to Dr. Terry Ackerman, whose kindness and wisdom drew me into this field. I would also like to extend my deepest gratitude to Dr. Bob Henson and Dr. John Willse for their genuine interest in their students' success, and for being superior models of deep curiosity. I am proud that all four agreed to serve on this dissertation committee.

The completion of this dissertation would not have been possible without the support of the National Commission on Certification of Physician Assistants (NCCPA). Thank you to Dawn Morton-Rias and Sheila Mauldin, who support the internship that allowed me to begin this line of research. I am especially grateful to Dr. Josh Goodman, a kind and deeply knowledgeable mentor and friend who saw merit in this project, as well as the other NCCPA psychometricians, Dr. Drew Dallas and Dr. Fen Fan, who offered advice and encouragement. Finally, my thanks to Reina Chau, whose first-rate programming made it possible to gather data from subject matter experts.

TABLE OF CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

CHAPTER I

INTRODUCTION

Context

Test developers know better than to hope that they may ever produce the perfect test instrument. Nevertheless, that academic, Platonic ideal serves a purpose: it is an aspiration, and its definition serves to point test developers toward best practices and around known obstacles.

The ideal form is one that presents the examinee with a series of questions or tasks, the responses to which give the examiner a reasonable snapshot of knowledge, skills, abilities, or other attributes of the examinee within a given domain. Perhaps this test instrument presents the examinee with a series of items that range in difficulty and address elements of the domain sufficient enough to capture a reliable picture. Every item in such an ideal scenario is equally unique among the items and sufficient in number to cover the breadth of the domain, and the examinee's answers to the questions tell us only what the examinee knows about the domain.

In practice, such instruments and items are truly Platonic: they are real and essential inasmuch as perfect joinery exists to a timber framer who can describe the essence of a perfectly fashioned mortise and tenon. And just as no timber framer has

ever fashioned the perfect joint, no item writer has written the perfect item, nor has any test developer assembled the quintessential test form.

In reality, of course, items do not work in perfect concert to measure latent traits among examinees. Some exams, out of necessity, are too short to provide robust information. For instance, to protect instructional time, a test instrument may be shortened significantly; in a fifth-grade classroom, a five-question quiz on multiplication is certainly going to be less reliable than a one-hundred-item test. This is a failure to meet the ideal by design.

In other cases, items (and the instruments that are comprised of them) fail to meet an ideal standard because they were written in such a way that they measure traits beyond those intended. Items like the following may introduce some form of bias and/or confounding information. For instance, an item may be so imbued with social baggage that it unintentionally measures something beyond the instrument's intent, such as family or social background. In the former SAT analogy section, you might find an item like this:

*bull:bear :: surge:*_____.

To answer this question successfully, the examinee must have *some* familiarity with the stock market. An item like this may express a degree of differential item functioning that manifests along the lines of class. In addition, the manner in which some exams are administered undermines the degree to which we may make arguments about their validity: it is inadvisable to use an electronic tablet to conduct a test of cognitive acuity amongst residents of a nursing home who are, perhaps, more likely to be in a heightened

state of confusion. Delivering the test or survey instrument in such an unfamiliar manner may well exacerbate that disorientation and confound the data that are being collected. These are examples of a failure to meet criteria of an ideal test instrument due to the introduction of bias.

In some cases, items can interact with one another in such a way as to provide a correct answer, reducing the useful information the test instrument is able to gather about an examinee. For instance, an item might read:

"What kind of pet did Annie have in the hit musical *Annie*?"

Imagine the examinee doesn't know the answer, but five minutes later she comes across the following question:

"What was the name of Annie's dog in the hit musical *Annie*?"

The second item *clues* the examinee to the correct answer to the first item—Annie has a pet dog—and, consequently, her correct response is a less useful measure of what the examinee knows about the hit musical *Annie* since it may be confounded with her ability to recall earlier items and test savviness.

Sometimes items interact in such a way as to simply distract the examinee. For instance, one item may cause the examinee to call into question their answer to another item. These two items are so similar they make candidates stop and say, "Wait. Something's wrong. I think I've already seen this." An example of these kinds of items is where the words between the two items are very similar, as may happen when stems are cloned. For instance:

A. An 8-year-old girl is brought to the emergency department by her parents because she has had a persistent headache for two days. On physical examination, the patient is found to be photophobic, and she is disoriented. Which of the following is the most likely cause of this patient's symptoms?

B. A 10-year-old boy is brought to the emergency department by his parents because he has had a persistent headache for two days. On physical examination, the patient is found to be phonophobic, and he is disoriented. Which of the following is the most appropriate treatment?

Where clueing is an example of items interacting in such a way as to make identifying the correct answer easier, in this scenario, the interaction between items might make the identification of correct answers more difficult.

Inattentive and test-savvy examinees alike may not be troubled by the similarity between these two items, but we can hardly fault other examinees who might stop and second guess how closely they have been reading items. We cannot fault them for wondering if they have already answered this item, nor can we fault them for wondering if, in the first item, they *thought* they saw the word "photophobic," but it could be that it *actually* read "phonophobic." Perhaps they will flip back to find the original and spend time identifying the differences between items, and perhaps they will begin to feel insecure about how accurately they have read other items. The items interact in such a way as to distract the examinee. The interaction deflects the examinee from the construct of interest.

Some items provide clues about the correct answers to other items, and some item pairs distract the examinee. Both types deflect focus from the construct of interest. A third variant of enemy interaction has the opposite effect: these items overemphasize a specific construct. Consider the following two hypothetical items[1]:

A. A three-year-old dog is brought to the emergency clinic, and the owners describe the onset of several unusual behaviors, including abnormal sound in barking, licking its own urine, abnormal licking of water, and regurgitation. Upon examination, drooping jaw and a dry drooping tongue are observed. What is the most likely diagnosis?

B. A five-year-old canine presents with symptoms that include biting and eating abnormal objects, biting with no provocation, running without apparent reason, stiffness upon running or walking, imbalance of gait, and frequent demonstration of the "dog sitting" position. What is the most likely diagnosis?

These items likely appear different to a lay audience, however anyone with veterinary training will recognize classic symptoms of rabies in dogs. As a matter of fact, both items test the very same content in much the same manner, though they use very different language to do so.

Item pairs such as the three examples above undermine the degree to which one may make a validity argument in favor of the instrument (Woo & Gorham, 2010). In addition, the instrument's capability to measure features of the construct itself—that is,

[1] The symptoms in these items were drawn from Tepsumethanon, Wilde, & Meslin (2005).

an examinee's knowledge of a given domain—is confounded by the interactions between items. These item pairs are known as enemy items, and test developers seek to remove them from test forms whenever possible.

It is incumbent upon test developers to assemble instruments that do not inadvertently introduce construct-irrelevant variance to test scores (American Educational Research Association, American Psychological Association, & National Council on Measurement in Education, 2014; Downing & Haladyna, 2013). One consideration in this effort is ensuring that enemies do not appear within the same form. As we see in the example of the two *Annie* questions, overlapping items within a form risk introducing inter-item dependencies, as a person would have an increased probability of responding either correctly or incorrectly to all the item enemies.

The inclusion of enemy items may also introduce other, more subtle, threats to score validity (Woo & Gorham, 2010). Similar items may not offer opportunities for clueing, but their resemblance may introduce confusion or dissonance, which undermines confidence in the outcomes of an exam.

Finally, the inclusion of enemy items can also introduce more severe threats to validity arguments. Indeed, inter-item dependencies, such as we would find when two items test the same thing, compromise measurement precision and negatively impact the degree to which scores might be considered valid (Woo & Gorham, 2010).

The most common approach for identifying enemies is manual review of item relationships by subject matter experts (SMEs) (Drasgow, Luecht, & Bennett, 2006). This process, however, can be expensive and cognitively taxing. Consider a scenario where

SMEs are asked to review forms with 200 items. The task requires that, as the SME makes her way through that form, she must hold in her working memory all items she has already read in order to perform pairwise comparisons. Even if one item is a near-identical clone of another, the SME may fail to identify the relationship, especially if the two items are distant from one another, as there are 19,900 item pairs on a 200-item form. She is less likely to notice that the 150th item is an enemy of the 5th item than she is that the 6th item is an enemy of the 5th. At the end of the day, having reviewed three forms, she has been asked to consider a whopping 59,700 item pairs.

Consider, too, the hourly pay-rate of a subject matter expert. From a fiscal perspective, not to mention that of common courtesy, it is prudent to make good use of their time and expertise. Where possible, the goal should be to reduce unnecessary cognitive loading and time spent on the identification of enemy items, when that time and expertise could be better spent on other elements of test construction and review.

In an ideal world, all item pairs in the item bank would be classified as either enemies or not enemies. An automated test assembly (ATA) algorithm could then simply avoid enemy pairs in the construction of test forms (van der Linden, 2005). However, it is one thing to review a form after it has been assembled; there are roughly 12.5 million unique item pairs in a bank of 5,000 items. Assuming on average one could classify an item pair every five seconds, this still constitutes nearly two years of uninterrupted work just for the review and coding of a modest item bank. More reasonably, the task is one that would require an army of SMEs. This gargantuan task is, of course, beyond cognitively taxing; it is fiscally and practically impossible.

Overview of this Approach

Because it is impractical to manually compare all item pairs in all but the smallest of banks, we must seek automated/algorithmic alternatives. There are latent characteristics that lie at the very heart of writing: while the combinations of words are overt and observed data, the meanings of those words are inherently latent. Capitalizing on computational resources, natural language processing (NLP) offers methods for identifying relationships among text documents. NLP is an area of study that lies at the intersection of computer science and linguistics and includes a number of fields, including text processing and summarization, speech recognition, and information retrieval (Chowdhury, 2003). Lai & Becker (2010), for instance, explored the feasibility of using NLP techniques (tokenization, stemming) paired with artificial neural networks (ANN) to identify enemy item pairs.

This study similarly uses items drawn from a national examination test bank, although the NLP data mining methodology is substantively different. In particular, this study embraces the topic modeling paradigm, wherein every document (an item in this context) is assumed to have been generated by a combination of latent topics. That is, one or more topics is assumed to contribute to any given text document; in the case of a very short document, like a test item, one might speculate that one or two topics meaningfully contribute to its composition.

The primary benefit of topic modeling in this context becomes obvious in light of two items that test the same concept but use different language to do so. (See the rabies example, above.) Although the words in the items are dissimilar, the *topics* are identical,

and this provides insight into the difficulty of detecting enemy pairs. While subject matter experts are very good at identifying these conflicts, lay editors and algorithmic approaches tend to be far less able to do so. Topic modeling may be the exception because it emphasizes the topics, not the words.

Similarity among items might be described, and indeed measured, according to the topics that contribute to their makeup. The probabilistic topic model at the heart of this study is latent Dirichlet allocation (LDA; Blei, Ng, & Jordan, 2003). LDA assumes that documents drawn from common latent topics will also draw from a collective pool of words. Those latent topics are uncovered by identifying groups of words in the text corpus that frequently occur together within documents. Furthermore, Blei, Ng, & Jordan (2003) note that documents have probability distributions over topics, and topics have probability distributions over words.

Other NLP techniques, such as latent semantic analysis, focus on word similarity between text documents. In a testing context, this approach is less desirable. Consider, for instance, a medical credentialing examination: one would expect a high degree of semantic similarity among items since they will tend to focus on the diagnosis and treatment of a relatively limited set of medical conditions. In addition, many testing organizations have codified the format of vignettes, insisting that information is presented in a specific order (e.g., age, gender, setting, symptoms, vital signs). This degree of similarity can cause NLP techniques that focus on word similarity to inappropriately flag enemy items at a higher rate. Because LDA focuses less on word similarity and more on the latent meaning underlying the words themselves, it is uniquely suited to identifying

enemy item pairs. LDA is an approach that is philosophically grounded in the understanding that the topics and ideas within text documents are what make them similar, and that there is a subset of words that are typically used to express those topics and ideas. In other words, "myocardial infarction" and "heart attack" are different at the word level, but they are identical at the topic level. LDA seeks to compare text documents in terms of the similarity between, or distance between, their topic probability distributions.

Topic modeling is a first step in identifying enemy pairs, as it can be used to describe the degree to which items are similar, or how close they are to one another, in terms of their topical makeup. This study takes a second critical step, however, and uses that measure of lexical proximity in conjunction with existing item metadata to classify item pairs as enemies or non-enemies. The metadata for the dataset at the heart of this study (an item bank used in medical credentialing) include 1) a list of item IDs that SMEs have determined are enemies of a given item, 2) the diagnosis on which the item focuses, 3) the International Classification of Diseases and Related Health Problems, Ninth Revision (ICD-9)[2] code with which the item is associated, the two blueprint dimensions—4) the Organ System and 5) Task—with which the item is affiliated, 6) the author of the item, and 7) the source from which the item was developed, where available.

[2] ICD systems are designed to be used for billing purposes, and the newest version is ICD-11. The dataset used in this study uses a legacy version since it is not used for billing, rather as a proxy for topic.

Using those metadata fields, including the LDA topic model output, a random forest model (Breiman, 2001) will be fit in order to classify item pairs as enemies or non-enemies. Random forest models are trained by fitting a number of decision trees to a sample of the complete set of observations and a sample of the variables/features. In the case of enemy items, the random forest model would be trained using a set of item pairs with a known enemy status. Imagine a rectangular dataset comprised of items by row and metadata (see previous paragraph) by column: a sample of rows and a sample of columns are used to determine the ideal classifier (decision tree) given that limited data. An ensemble of those classifiers is then assembled to train the Random Forest model using known enemy pairs. The result is a likelihood that any two items are enemies. Using that likelihood, one can set a cut off and just say you're not going to use any items that have a 90% or higher likelihood of being an enemy pair.

Random forests are ideal for this use case because they are flexible, stable, and, since they are intentionally developed using incomplete data (Breiman, 2001), robust in the presence of missing data. These models work well with continuous data and categorical data alike. This means continuous data, such as the LDA-generated distance between two items, as well as categorical metadata, such as authorship, may be used simultaneously to train the model and to ultimately classify item pairs as enemies or non-enemies.

It is the intent of this study to determine whether an LDA-Random Forest approach such as this can identify enemy item pairs in an operational bank with sufficient degrees of accuracy, specificity, and sensitivity. In the following chapter, the motivation

for this study will be provided within a context of prior research into the issues and prior approaches.

CHAPTER II

LITERATURE REVIEW

This chapter seeks to provide an overview of enemy items in the literature. Definitions of enemy items relationships will be explored, and justifications for identifying and avoiding enemy pairs will be discussed. Specifically, concerns about validity arguments will be discussed, including concerns that are entangled with issues of measurement precision and item response theory assumptions. Finally, this chapter will discuss methods of identifying enemy item pairs, from content expert form review, to making educated guesses, to post hoc statistical approaches, to natural language processing approaches that are now being developed.

Enemy Items

Enemy items are item pairs that have "characteristics that are so similar in content […] that in most cases, the items would not be administered together on one test to the same examinee" (Woo & Gorham, 2010, p. 15). Some item pairs are enemies because the stems and answer options are, in fact, duplicates of one another, or the stems alone are duplicates or near duplicates. Other item pairs are considered enemies because the content overlaps in such a way as to test the same construct (Case & Swanson, 1998; Downing & Haladyna, 2006; Haladyna & Downing, 2004), or in such a way that an interaction between the items points the examinee toward the correct answer, which is known as "clueing" (Woo & Gorham, 2010) or "cueing" (Ackerman & Spray, 1986).

Degrees of duplication, as described above, can arise as artifacts of the item development process. Surface duplication, for instance, may arise when item writers take one well-behaving item and modify it to create a new item that could be used on a parallel form. Taken even further, within an automatic item generation (AIG) context (Gierl & Haladyna, 2013), so-called *parent* items may be cloned, whereby surface features of the parent item are altered, creating an *item shell* (Haladyna & Shindoll, 1989) and, ultimately, a family of similar items. The identification of enemy items is inherently an effort to improve the degree to which a test instrument might be considered valid.

Validity Arguments

While there has been little discussion in the literature about the why enemy items should be eliminated from test forms, one consequence of their inclusion is that they may undermine the validity arguments one may make in support of the intended uses of the instrument, a subject about which a great deal has been written.

Validity is the degree to which evidence and theory support the intended interpretation and use of a test score (American Educational Research Association et al., 2014). Kane (1992, 2004, 2006, 2013) argues that validation is really a two-step process. First, those who build and administer test instruments must make a clear interpretation and use argument (IUA): that is, the intent of the instrument must be explicit. Second, they must endeavor to evaluate the degree to which the administration of the exam instrument(s) supports the IUA. The evaluation of those claims, Kane (2006) suggests, is undertaken by evaluating the assumptions one makes about 1) scoring (generating a score to assess performance), 2) generalization (the overall quality of the test instrument), 3)

extrapolation (the conclusions we draw about real-world performance based on test performance), and 4) implications (how the test results will be used). In the context of the present study, construct underrepresentation and irrelevant test variance are two threats to these assumptions that ought to be considered (Cook & Campbell, 1979; Kane, 2006; Messick, 1989).

Enemy pairs most often cover closely overlapping material, often the exact same content. A typical enemy item pair, consequently, may interrogate the domain of interest redundantly, or half as much as was intended. Validity arguments may be undercut when the instrument is intended to measure a given trait or domain of knowledge, but the items do not sufficiently cover the construct to do so reliably or in a generalizable manner. This is known as construct underrepresentation, or construct deficiency (American Educational Research Association et al., 2014).

Sometimes an examinee may arrive at a correct answer by some means other than mastery of the material at hand. For instance, an examinee who has mastered the material in question may not be able to identify the correct answer option if the stem is written in such a way as to make its meaning inscrutable or confusing. One common example is that of examinations delivered in a language in which the examinee is not fluent. Conversely, when none of the distractors are in any way distracting, the examinee need know nothing about the construct of interest in order to select the one obvious correct answer option. This is known as construct irrelevant variance, or construct contamination (American Educational Research Association et al., 2014).

For instance, when enemy item pairs interact in such a way as to induce confusion or cognitive dissonance, construct irrelevant factors are also at play. Woo & Gorham (2010) indicate that identical stimuli (such as duplicate figures, tables, or passages) across items that test different content on the same form may, at the very least, be distracting:

> This category describes a fairly common practice for large-scale item production. Supplemental item stimuli such as custom-produced graphics, exhibits, custom sounds, or costly copyrighted reading passages or other elements can be reused in multiple items to reduce cost and to make the best use of staff resources for research and collection of stimuli elements. […] In a pretest setting in which groups of items using the same stimuli elements might be pretested together, this issue becomes even more important to acknowledge and to manage. For example, an examinee who receives three pretest items on one test, all using the same graphic stimulus, will likely become confused and overwhelmed from seeing the same graphic in three different items. This is especially true in computer-based testing, where candidates are not allowed to review items that have already been answered. (p. 16)

Items with such features distract the examinee and make the exam unnecessarily and unintentionally more difficult. This is similar to making an item more difficult by making the language more difficult to parse. Where clueing is an example of items interacting in such a way as to make identifying the correct answer easier, in this case the interaction between items might make the identification of correct answers more difficult. The most basic example of such replication is as follows:

A) Which of the following environmental pollutants have been shown to contribute to cardiovascular disease?

B) Which of the following environmental pollutants have not been shown to contribute to cardiovascular disease?

Inattentive or test-savvy examinees alike may not be troubled by the similarity between these two items, but we can hardly fault other examinees who might stop in their tracks and second guess how closely they have been reading items. We cannot fault them for wondering if they have already answered this item. Perhaps they will flip back to find the original question and spend time identifying the difference between items, and perhaps they will begin to feel insecure about how accurately they have read other items. The items interact in such a way as to deflect the examinee from the construct of interest.

Items that clue examinees to the correct answers to subsequent questions undermine the validity of an instrument, as those items do not measure the intended construct exclusively. Indeed, in cases where the examinee is alerted to a correct answer by a previous question, the item measures pattern recognition or test savviness, and perhaps measures nothing whatsoever to do with the intended construct. For instance, the following two items should not appear on the same form:

A) Environmental pollutants, such as tobacco smoke, have been shown to contribute to which one of the following types of disease? (Answer: Cardiovascular disease)

B) Which one of the following environmental pollutants has been shown to contribute to cardiovascular disease? (Answer: Tobacco smoke)

Whereas examinees who do not recognize the relationship between the items *are* providing information about the domain of interest, in aggregate this information is confounded by the former group of examinees.

In both of the above cases, the deflection of focus from the construct of interest undermines the validity of the instrument, and the ability to measure features of the construct itself—an examinees knowledge of a given domain—is confounded by the interactions between items.

Considerations of IRT and ATA

Not unrelated to the above discussion of validity arguments, there are issues of related to item response theory (IRT) is used to evaluate test items, to construct test forms, and to score examinee responses. Underlying IRT are two assumptions with which this study is particularly concerned. First, one assumes that all items are unidimensional; each item must measure only one trait, such as the ability to solve single digit addition problems (Embretson & Reise, 2000; Lord & Novick, 1968). When examinees correctly answer an item because they recognize that one items offers a clue to the correct response on another, the test instrument no longer measures the intended trait, but it measures an additional trait that we might call test savviness. Furthermore, if two test items are phrased so similarly yet still measure discrete traits, the similarity in stems may be enough to introduce some degree of cognitive dissonance that may induce the examinee to balk or question her intuition about the construct. This potentially introduces construct irrelevant variance. Whereas clueing may artificially drive scores up, this latter example of cognitive dissonance may drive scores downward while telling us nothing about the trait in question. Examinees are not computers—they remember previous items, which may help them or hurt them. Either way, the relationships between items can introduce dimensionality beyond what was intended.

Second, IRT operates under the assumption that test items are locally independent, that one item does not influence the response to another; rather, an examinee's response is predicated on her trait levels (Embretson & Reise, 2000). Put another way, in a unidimensional model, a single latent trait of interest ought to explain response patterns, however, when another trait influences those patterns, multidimensionality is present (Mellenbergh, 1994). In the case of clueing (Woo & Gorham, 2010), variance attributable to a latent trait is confounded by an interaction between items that may have no more to do with the examinee than whether she recognizes that one item provides the answer to another item. Likewise, in the case of two items that test the very same content, we would expect to see covariance that is rooted in the enemy relationship.

Traditional and Novel Approaches to Identifying Enemy Items

Limited discussion of enemy items can be found throughout the literature on test assembly. Overwhelmingly, however, those instances are limited to a few sentences indicating that enemy item pairs should not be included on the same form (Drasgow, Luecht, & Bennett, 2006; Huitzing, Veldkamp, & Verschoor, 2005; Luecht & Sireci, 2011; van der Linden, 2005; Veldkamp, 2013), though very few discuss why this is. Fewer still discuss the processes by which enemy item pairs may be identified, relegating these efforts, perhaps, to the realm of editorial labor.

Indeed, the vast majority of enemy item pairs are identified in a joint *a priori* endeavor between editors and subject matter experts before forms are published (Lai & Becker, 2010; Woo & Gorham, 2010). During form review, complete exam forms are

distributed to content experts who pore over the items looking for surface errors, outdated content, mis-keyed answers, and enemy pairs. The latter task is in many ways the most difficult and prone to error, as the task requires a form review committee member to conduct a pairwise comparison of, in many cases, hundreds of items. The task can take hours, sometimes spanning more than a day.

Others have suggested statistical approaches for identifying enemy item pairs. Ackerman & Spray (1986) proposed a general model for item dependency, one benefit of which was the possibility of detecting enemy items by exposing violations of local independence caused by cueing. Still others have suggested using Yen's Q_3 statistic (Goodman, 2008; Yen, 1984, 1993) to identify local item dependencies that may indicate enemy item interactions (Pommerich & Segall, 2008). One drawback of such approaches, however, is that they are *post hoc* in nature: in order to see evidence of local dependencies, the exam form must have been administered to a sufficient number of examinees. On one hand we might argue, "better late than never." On the other hand, however, the proverbial horse has left the testing center.

An efficient, rigorous, *a priori* method of identifying enemy item pairs is the goal. Natural language processing (NLP) offers some tools for identifying such pairs. Becker & Kao (2009) use the cosine similarity index (CSI)—a measure of the similarity between two text documents—to try to identify potentially stolen items, as well as enemy item pairs. Lai & Becker (2010) use three similarity indices—CSI, unit-overlap, and longest common subsequence (LCS)—in conjunction with item metadata to train a predictive artificial neural network (ANN) to identify enemy item pairs. Inconsistent results, the

authors note, were likely due in part to a small sample of items (N = 266). In addition, they note that the similarity indices they used were oriented toward syntactic features; that is, the surface-level feature of two texts did not provided enough meaningful information to allow the ANN more consistently identify enemy pairs. The authors suggest that future studies should use a larger sample of items, should be more intentional with regard to developing training sets, and seek similarity indices that emphasize semantic features at least as much as syntactic features.

This study proposes to do just that by leveraging two machine learning approaches. Topic modeling (Blei et al., 2003; Griffiths & Steyvers, 2004; Papadimitriou, Raghavan, Tamaki, & Vempala, 2000) is a class of machine learning and natural language processing approaches that seeks to measure the semantic similarities between text documents. Using a similarity index provided by topic modeling, combined with item metadata, a random forest model (Breiman, 2001b) will be used to classify enemy item pairs.

Topic Modeling and Latent Dirichlet Allocation

Machine learning is a novel paradigm that stands in contrast to other algorithmic approaches. Traditionally, data is provided as input, and via an algorithm, output is generated. For instance, response pattern data for a given exam may be provided to an item response theory algorithm of one form or another, and in the output we find a distribution of scores. By contrast, instead of providing a formula, the machine learning algorithm is tasked with deriving that formula. In a supervised machine learning algorithm, we provide not only the input data, but also some *output* data, and we ask the

algorithm to arrive at a formula which bridges the two. By contrast, in unsupervised machine learning approach, the model is provided *only* the input data and is tasked with finding patterns therein. Machine learning approaches are particularly good at identifying patterns in large structured and unstructured data sets, allowing for the clustering of observations, classification of observations, and predictive analysis based on those patterns.

Natural language processing (NLP) is a field of study concerned with the use of computers to find meaningful patterns in natural language (Chowdhury, 2003). NLP algorithms are ubiquitous today: search engines provide suggestions about intended or similar queries as we type; the same search engines serve results based on what they think the user meant; powerful applications found on telephones recognize speech, can differentiate between voices and speech patterns, can create reasonable transcripts in real time, and can even translate spoken words in to other languages in real time.

Topic modeling is a natural language processing (NLP) machine learning approach that is intended to identify patterns in unstructured text data. Topic Modeling was developed primarily for its utility in data retrieval; related documents can be identified according to their topical similarity. For instance, "favoriting" an article that was read online may prompt the host site to provide a list of similar or related articles; topic models are well suited for identifying these similarities among documents. In fact, topic models are not limited to text document analysis; topic modeling approaches are used to identify similarities among all manner of documents, including images.

Latent Dirichlet Allocation (LDA; Blei et al., 2003) is a topic model in the unsupervised machine learning paradigm. LDA is an approach that estimates the contribution of latent topics to a given document. Whereas a clustering approach, such as *k*-means, would assign clusters of documents to latent topics, the LDA approach assumes that any given topic—indeed, any combination of topics—may contribute to a given document, and this is expressed as a distribution of probabilities across all possible topics. While LDA is used to uncover latent topics from a text corpus (a collection of text documents), the way in which it identifies those topics lends itself to document comparison: documents can be compared in terms of the similarity between, or distance between, their topic probability distributions.

Overview of LDA

In the context of LDA, a word is a basic unit of discrete data from a vocabulary indexed by $\{1,...,V\}$. Words are represented as unit-basis vectors: that is, each word w is represented by a vector of length V and $w^v = 1$ and $w^u = 0$ where $u \neq v$. A document, denoted by $\mathbf{w} = (w_1, w_2, \ldots, w_N)$, is a group of N words. Note that LDA is a *bag-of-words* model, and as such the order of the words is insignificant beyond simple indexing of the word. Finally, a corpus, denoted by $D = \{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_M\}$, is a group of M documents.

A document $\mathbf{w}_M$, as described by Blei, Ng, & Jordan (2003), is generated in the following manner: A priori, the number of topics (*k*) is fixed. The generative process for a document $\mathbf{w} = (w_1, ..., w_N)$ of a corpus *D* containing *N* words from a vocabulary

consisting of V unique terms, $w_i \in \{1,...,V\}$ for all $n = 1,...,N$, consists of the following three steps:

1. Choose $N \sim \text{Poisson}(\xi)$.
2. Choose $\theta \sim \text{Dir}(\alpha)$.
3. For each of the N words w_n:
    a. Choose a topic $z_n \sim \text{Multinomial}(\theta)$;
    b. Choose a word w_n from a multinomial probability distribution conditioned on the topic z_n: $p(w \mid z_n, \beta)$.

The number of topics k is assumed to be known a priori; this is a limitation of the LDA framework that will be discussed later in this chapter. The probability distributions of words as generated by topics can be found in a β-matrix with dimensions $k \times V$, where $\beta_{ij} = p(w^j = 1 \mid z^i = 1)$; an example of an empirical β-matrix in transposed, $V \times k$ orientation can be found in Appendix A.

The number of words, N, in a document is given by

$$p(x;\lambda) = \frac{\lambda^x e^{-\lambda}}{x!} \tag{1}$$

where λ is the expected value, or the total number of words in the corpus over the total number of documents in the corpus, and where x is the object of interest. Blei, et. al. note that the Poisson distribution is not critical and other document length distributions can be used. The topic distribution for a given document is given as

$$p(\theta \mid \alpha) = \frac{\Gamma\left(\sum_{i=1}^{k} \alpha_i\right)}{\prod_{i=1}^{k} \Gamma(\alpha_i)} \theta_1^{\alpha_1 - 1} \ldots \theta_k^{\alpha_k - 1} \tag{2}$$

where α is a vector of length k of positive real numbers drawn from a Dirichlet distribution, typically with values of less than 1; lower values of α indicate that fewer topics contribute to a given document under this model. An example of the resulting topic distributions can be found in the $D \times k$ matrix in Appendix C.

The joint distribution of a topic mixture θ, a set of topics $\boldsymbol{z}$, and a set of words $\mathbf{w}$, given the parameters α and β, is expressed by

$$p(\theta, \mathbf{z}, \mathbf{w} \mid \alpha, \beta) = p(\theta \mid \alpha) \prod_{n=1}^{N} p(z_n \mid \theta)\, p(w_n \mid z_n, \beta). \tag{3}$$

The marginal distribution of a single document is found as follows:

$$p(\mathbf{w} \mid \alpha, \beta) = \int p(\theta \mid \alpha) \left(\prod_{n=1}^{N} \sum_{z_n} p(z_n \mid \theta)\, p(w_n \mid z_n, \beta) \right) d\theta. \tag{4}$$

The product of the above marginal probabilities (Eq. 4) provides the probability of a corpus:

$$p(D \mid \alpha, \beta) = \prod_{d=1}^{M} \int p(\theta \mid \alpha) \left(\prod_{n=1}^{N} \sum_{z_{dn}} p(z_{dn} \mid \theta_d)\, p(w_{dn} \mid z_{dn}, \beta) \right) d\theta_d \ . \tag{5}$$

Because the computation of posterior distributions of the of the hidden variables in a given document (α, β, θ) is intractable, Blei et al. (2003) proposed a modified expectation-maximization algorithm, variational expectation-maximization (VEM). In the E-step, the optimizing values of the variational parameters are found for each document; those parameters are the Dirichlet parameter γ and the multinomial parameters $(\phi_1, \ldots, \phi_N)$, where $\{\gamma_d^*, \phi_d^* : d \in D\}$. Variational parameter values are optimized by minimizing the Kulback-Leibler divergence between the variational distribution and the true posterior. In the M-step, the resulting lower bound on the log likelihood with respect to α and β is maximized.

Discussion of LDA

LDA assumes that documents drawn from common latent topics will also draw from a collective pool of words. Those latent topics are uncovered by identifying groups of words in the text corpus that frequently occur together within documents. Furthermore, Blei, Ng, & Jordan (2003) note that documents have probability distributions over topics, and topics have probability distributions over words. This emphasis on probability distributions rather than on strict word frequencies is in part what differentiates LDA from other topic modeling approaches, such as latent semantic indexing (LSI; Deerwester, Dumais, Furnas, Landauer, & Harshman, 1990; Papadimitriou et al., 2000), also known as latent semantic analysis (LSA).

Plate notation, a graphical representation of iterative steps, in conjunction with a description of the generative process, offers a concise overview of LDA. In the plate

notation on the following page (Figure 1), plate *M* represents the total number of documents within a text corpus (i.e. a body of discrete documents), and plate *N* represents the words within each document. A high value Dirichlet prior for topic distributions (α) indicates that any given document will exhibit the influence of a higher number of topics, whereas a low value for α indicates that only a handful of documents will contribute meaningfully to any given document. The α value is very much dependent on the text corpus itself. That is, if the documents under consideration (i.e. the text corpus) are by nature highly focused (perhaps due to brevity and convention), then a low α value makes sense. For example, a text corpus of tweets, because they are capped at 140 to 280 characters, might benefit from a low starting value for α. Similarly, a high value for the Dirichlet prior on the per-topic word distribution, β, suggests that each topic is comprised of a mixture of most of the words, while a low β, indicates that topics are likely to be comprised of a mixture of only a small proportion of the words in the text corpus. The topic distribution for a given document *M* is denoted as θ_m. Each topic is denoted as z_{mn}, the topic for the n^{th} word in document *m,* and the w_{mn}, the word itself.

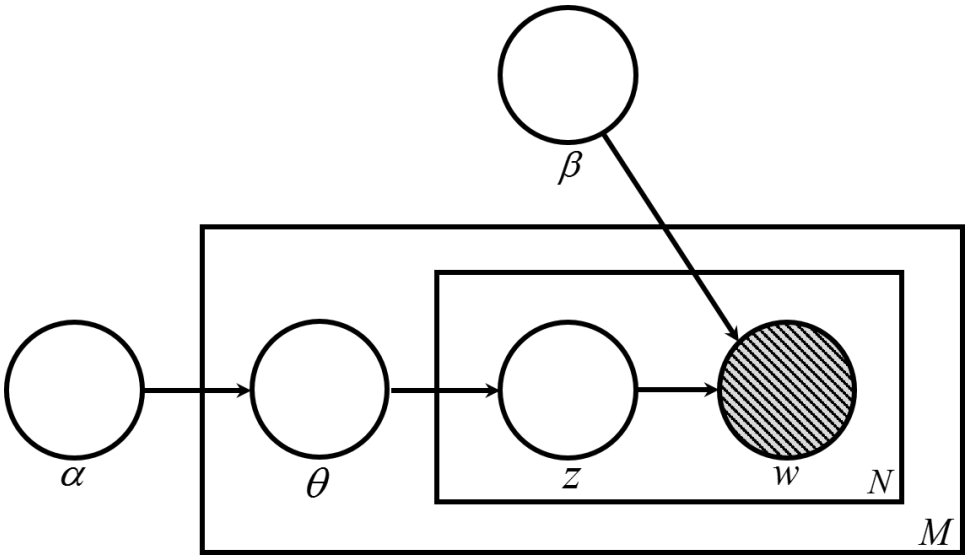


Figure 1. Plate Notation for the LDA Model. Plates represent repeated choices: plate *N* represents the reoccurring choice of topics and words within a document, and plate *M* represents the documents within the corpus. Hidden nodes are unshaded, and observed nodes are shaded.

Latent Dirichlet Allocation assumes that documents are generated in the following manner: First, the number of words for a given document is established, then the topic distribution is determined; if there are five topics, then the proportional contribution from each topic to the document is estimated (eg. Topic A: 5%, Topic B: 10%, Topic C: 45%, Topic D: 30%, Topic E: 5%). Finally, words are chosen for the document by selecting a topic based on the multinomial distribution of topics, determined in the previous step, and then selecting a word based on the multinomial distribution of words for that topic.

This generative process, of course, is not at all how texts are created; nevertheless, understanding this generative framework provides insight into how the LDA algorithm operates, and it allows for stronger intuition with regard to the model's strengths and weaknesses. The algorithm itself works backwards through the generative process.

Given a corpus of *M* documents, the LDA algorithm will learn the topic distribution of *K* topics for each document, as well as the word distribution for each topic. This is achieved in the following manner: First, the algorithm randomly assigns each word in each document to one of the *K* topics. For each document, it assumes that all of the topic assignments are correct with the exception of the current one, and then it calculates two proportions. First, the proportion of words in the document that are currently assigned to the topic is calculated: $t = p(topic\ t|document\ d)$. Second, the proportion of assignment to topic *t* over all documents that come from this word is calculated: $w = p(word\ w \mid topic\ t)$. Finally, the product of the two probabilities is used to assign the word to a new topic: $p(topic\ t|document\ d)\ \times\ p(word\ w \mid topic\ t)$.

More formally, the LDA framework operates under the assumption that documents are generated in the following manner: For text corpus *D* consisting of *M* documents, each of length N_i, choose $N \sim Poisson(\xi)$, choose $\theta \sim Dir(\alpha)$, and for each of the *N* words (w_n) choose a topic $z_n \sim Multinomial(\theta)$ and a word (w_n) from $p(w_n|z_n, \beta)$, a multinomial probability conditioned on the topic z_n. This generative model reflects the intuition that documents are informed by multiple topics: indeed, each document is "generated" by a unique proportion of the topics, and every word in the document is drawn from one of those topics (Blei, 2012).

Topic Distributions and Document Similarity

The topic probability distributions across documents (Equation 2; Appendix C) estimated by the LDA model provide a means for calculating the similarity between documents. Commonly, similarity between probability distributions is assessed by

calculating using Kulback-Leibler divergence. The Kulback-Leibler divergence of Q to P is expressed as:

$$D_{KL}(Q \parallel P) = \sum_i Q(i) \ln\left(\frac{Q(i)}{P(i)}\right) \tag{6}$$

In machine learning, Kullback-Leibler divergence is understood as the information gain attained if P is used instead of Q. The calculation is not symmetrical, however: $D_{KL}(Q \parallel P) \neq D_{KL}(P \parallel Q)$. In the context of topic modeling and document similarity, this behavior does not make sense, as the divergence of the $D \times k$ probability distribution for document $\mathbf{w}_1$ to the probability distribution for document $\mathbf{w}_2$ *should* be symmetrical: that is, if $\mathbf{w}_1$ is similar to $\mathbf{w}_2$ in terms of how topics are represented, then $\mathbf{w}_2$ is should be *equally* similar to $\mathbf{w}_1$.

The Jensen-Shannon divergence, a variant of Kulback-Leibler divergence, accounts for this requirement in topic modeling, as it is symmetrical. Symmetry is achieved by calculating a midpoint, or average probability distribution, and then calculating the Kulback-Leibler divergence of each probability to that midpoint:

$$D_{JS}(Q \parallel P) = \frac{1}{2} D_{KL}(Q \parallel M) + \frac{1}{2} D_{KL}(P \parallel M) \tag{7}$$

where $M = \frac{1}{2}(Q + P)$. Now, two texts may be compared in terms of divergence, and the order of comparison does not matter. In addition, because the condition of symmetry has

been met, the Jensen-Shannon divergence can be converted to a metric, the Jensen-Shannon distance (JSD): $\sqrt{D_{JS}(Q \| P)}$. While it is a metric of distance, it is perhaps more intuitive in the context of topic modeling to consider the JSD a metric of topical similarity between two documents.

Stop Words and Stemming

It is common practice to remove words that occur so frequently as to render them uninformative, such as "the," "if," "but," and "and." These words are known as *stop words,* and we define them as terms that have just as likely to be found in documents that are related or similar as they are to be found in completely unrelated documents (Wilbur & Sirotkin, 1992). In the context of a search string in a document query, these are the words that are unlikely to These words tend to be more syntactic than semantic, and as such, their inclusion in a text corpus serves only to introduce noise to analyses: garbage in, garbage out. Removing stop words reduces the text corpus to only words that do the real work of communicating ideas. It should be noted that stop words are not limited to the prepositions, conjunctions and articles listed above. Indeed, when a text corpus is focused on a single topic or group of related topics, such as one might find in medical literature, certain words become so common as to provide less meaning than they would otherwise in other contexts. In medical literature, "patient," "physical," "blood," and "examination" might be flagged as stop words and filtered from a dataset.

How should stop words be removed? Term frequency is one way to think about word importance, or unimportance, as discussed above. More frequent terms tend to be less informative, while more infrequent words can provide more clues about topic

membership. However, reliance on term frequency alone is not enough. Rajarman & Ullman (2011) note that some words, such as "albeit" and "notwithstanding," while uncommon, are not particularly informative.

Similarly, it is common practice to remove suffixes and sometimes prefixes so that root words are more accurately represented in the text corpus. This process is called *stemming*. For instance, to a topic modeling algorithm, walk, walk***ed***, walk***ing***, walk***er,*** and walk***s*** are five distinct terms, but NLP approaches like topic modeling typically benefit from reducing the variants to a common base form: walk. Lemmatization is the process by which words are reduced to their roots as found in a dictionary; this process is often aided by lookup tables for difficult word reductions, such as "saw" → "see". Stemming, by contrast, is an algorithmic approaches that seeks to reduce words to a base form. The Porter stemmer (Porter, 1980) assess each word in the sample, working from the final character in the string to the first. The end of each word is scanned for suffix patterns; when a pattern is identified, characters are either replaced or deleted depending on the stemming rule. For instance, whenever the algorithm finds the *-sses* suffix, it removes *-ss*, so "assesses" → "assess", "bosses" → "boss", and "stresses" → "stress".

Unlike lemmatization, in many instances stemming will produce base forms that will not be found in a dictionary. For instance, the Porter stemmer removes the suffixes *-al, -ism,* and *-ize*, so "communism," "communal," and "communize" all become "commun." While "commun" cannot be found in a dictionary, it serves as a more informative word in a topic modeling context than its discrete variants. Concentrated in

this manner, an LDA model is more able to estimate a topic around or including communism.

Number of Topics

LDA models are predicated on the number of topics that are thought to be responsible for the generation of all documents in the text corpus. The number of topics must be specified before fitting and LDA model. While there is no one best way to determine the ideal number of latent topics, some common sense can point the researcher in the right direction. For instance, if the text corpus is both large and wide ranging in terms of content—think Wikipedia—then it is likely that a very large number of topics have contributed to the documents therein. But how many topics? The model requires a number, and it is difficult to get from *a lot* or *a few* to a specific, reasonable value.

Perplexity, perhaps the best most widely used fit index for topic models, and for language models more generally (Blei et al., 2003), asses the likelihood of a word appearing in a given document. The LDA is fit to a training set, and then perplexity asses fit using a held-out validation set. For a test of M documents, perplexity is given as:

$$perplexity\left(D_{val}\right) = \exp\left\{-\frac{\sum_{d=1}^{M}\log p\left(\mathbf{w}_d\right)}{\sum_{d=1}^{M}N_d}\right\} \tag{8}$$

which is a decreasing function of the log-likelihood of the unseen documents. Lower values for perplexity indicate a better fit of the model.

Another approach for determining a reasonable number of topics for a given model and corpus is to look for evidence of topic structure stability. When too few topics

are specified, pairs of topics may correlate strongly (Cao, Xia, Li, Zhang, & Tang, 2009), the effect of which is that some number words in a document may be associated roughly equivalently with multiple topics. The average cosine distance between every pair of topics serves as a measure of topic structure stability:

$$dist_{avg} = \frac{\sum_{i=0}^{K} \sum_{j=i+1}^{K} corre\left(T_i, T_j\right)}{K \times \left(K-1\right)/2} \tag{9}$$

Using the average cosine distance as a radius r about a topic Z, the number of topics within the radius of r from Z is the density of Z: $Density\left(Z, r\right)$. Cao et al. suggest that selecting the appropriate number of topics based on topic density has two effects: similarity will be as large as possible within topics and the topics may represent more explicit meaning, and similarity will be as small as possible across topics, allowing for a more stable structure.

Arun, Suresh, Veni Madhavan, & Narasimha Murthy (2010) found that the singular value distribution of the topic- word matrix ($k \times w$; Appendix A) is close to the distribution over the row L_2 norm (least squares error) of the document-topic matrix ($D \times k$; Appendix C) when topics become orthogonal. Because distributions over L_2 norms and L_1 norms tend to converge when for random matrices in high dimension, they become comparable. Given these two observations, the authors propose measure of fit that compares the singular value distribution of the topic- word matrix ($k \times w$; Appendix

A) with the L_1 norm (least absolute deviations) of the document-topic matrix ($D \times k$; Appendix C).

These three indices can be used to assess the fit of several potential LDA models across a range of specified topics. In aggregate, these indices provide a defensible range of the appropriate number of topics for a corpus.

Random Forest Classification

Latent Dirichlet allocation (LDA) models are unsupervised machine learning models. That is, the model is not trained on a set of data that has already been labeled or classified (Jain, 2010). Random forest classifiers, on the other hand, are supervised models, because they are trained and validated first on data for which the classification of cases is known. In the case of enemy item detection, a training set of known enemy and non-enemy pairs would be used to train the model. In this use case, the random forest is provided a training set of data that includes items that have already been identified by SMEs as enemies as well as items that have not been classified as such.

In addition to being supervised, random forest models are members a class of algorithms known as ensemble systems (Zhang & Ma, 2012). Ensemble systems combine the results of multiple algorithms in an effort to reduce variance and increase our confidence in the results (Polikar, 2012). The appeal of ensemble methods is intuitive: the task of choosing elected officials in a democratic fashion is an ensemble method. Likewise, convening a standard setting panel of subject matter experts to set a cut score is an ensemble method. Random forests are an example of this method, as they are an ensemble of many decision trees (hence, "forest"). Decision trees are relatively

easy to construct and they produce models that are easy to interpret. That is, a single decision tree that is fit to a given training set will typically be easy to interpret given a reasonable familiarity with the data. However, Hastie, Tibshirani & Friedman (2009) note that "[decision trees] have one aspect that prevents them from being the ideal tool for predictive learning, namely inaccuracy," (p. 352). Individual decision trees tend to overfit the data considerably.

Random forest models seek to circumvent this shortcoming by following a *bagging* approach. Bagging (Breiman, 1996), which is derived from **B**ootstrap **Agg**regation, attempts to reduce variance by randomly sampling with replacement from the training dataset and then building decision trees based on each of those bootstrapped datasets. In a classification context, such as the classification of an item pair's enemy state, the mode of the decision tree outputs is used (Zhou, 2012).

Bagging in the context of classification is understood to function in the following manner: for a training set of T consisting of data $\{(y_n, \boldsymbol{x}_n), n = 1, \ldots, N\}$ where y is a categorical value, a procedure is used to generate a predictor $\varphi(\boldsymbol{x}, T)$, where the input is $\boldsymbol{x}$ and the variable to be predicted is y. Take repeated bootstrap samples $\{T^{(B)}\}$ from T and form $\{\varphi(\boldsymbol{x}, T^{(B)})\}$ and let it vote to form $_{\varphi B}(\boldsymbol{x})$.

Under this framework, a dataset is randomly divided into a training set T and a validation set V. From T a bootstrap sample of T_B is selected and a decision tree is grown from it. This is repeated *N* times, providing tree classifiers $\phi_1(x), \ldots, \phi_n(x)$. Data from *V* are passed through the decision trees, and the classification with the most "votes"

in $\phi_1(x),\ldots,\phi_n(x)$ (i.e. the plurality of classifications) is the estimated class. The misclassification bagging rate, $e_B(T,V)$, proportion of times the estimated class differed from the true classification, or the error rate: $\frac{FP+FN}{P+N}$.

The random forest model is rooted in bagging, but it adds the following step to the algorithm: when training each decision tree, at each split the model selects a random subset of independent variables, or features, from the training set.

More formally, the algorithm for random forest classifiers from Hastie et al., (2009) is as follows:

1. For $b = 1$ to B:
    a. Draw a bootstrap sample T^* of size N from the training data.
    b. Grow a random-forest tree T_b to the bootstrapped data, by recursively repeating the following steps for each terminal node of the tree, until the minimum node size n_{min} is reached.
        i. Select m variables at random from the p variables.
        ii. Pick the best variable/split-point among the m.
        iii. Split the node into two child nodes.
2. Output the ensemble of trees $\{T_b\}_1^B$.

In the same fashion as described for the bagging approach, $\hat{C}_b(x)$ is the classification estimate for the bth random forest tree, and the mode of $\left\{\hat{C}_b(x)\right\}_1^B$ becomes the classifier $\hat{C}_{rf}^B(x)$ (p, 588).

Consider the following simplified training set consisting of six item pairs:

Table 1. Example Training Set

Item A ID	**Item B ID**	**Blueprint, Item A**	**Blueprint, Item B**	**Jensen-Shannon Distance**	**Enemies**
1	2	Cardiovascular	Cardiovascular	0.01	Yes
1	3	Cardiovascular	Cardiovascular	0.52	No
1	4	Cardiovascular	Infectious Diseases	0.31	No
2	3	Cardiovascular	Cardiovascular	0.77	No
2	4	Cardiovascular	Infectious Diseases	0.02	No
3	4	Cardiovascular	Infectious Diseases	0.02	Yes

Item ID A and Item ID B are identifiers, Blueprint, Item A and Blueprint, Item B are independent categorical variables related to the test blueprint, Jensen-Shannon Distance is an independent continuous variable that indicates the topical similarity between the two items (from LDA), and Enemies is a dependent categorical variable that indicates whether or not subject matter experts have classified these items as an enemy pair.

To create the bootstrap sample, the algorithm will draw, with replacement, from the training set, perhaps arriving at the following sample:

Table 2. Example Bootstrap Set

Item A ID	Item B ID	Blueprint, Item A	Blueprint, Item B	Jensen-Shannon Distance	Enemies
1	2	Cardiovascular	Cardiovascular	0.01	Yes
1	2	Cardiovascular	Cardiovascular	0.01	Yes
1	2	Cardiovascular	Cardiovascular	0.01	Yes
2	3	Cardiovascular	Cardiovascular	0.77	No
3	4	Cardiovascular	Infectious Diseases	0.02	Yes
3	4	Cardiovascular	Infectious Diseases	0.02	Yes

Next, a series of decision trees will be grown using a sample of the independent variables. The first sample of independent variables might consist of Blueprint, Item A and the Jensen-Shannon Distance; the second sample might consist of Blueprint, Item B and the Jensen-Shannon Distance; and the third sample might consist of Blueprint, Item A and Blueprint, Item B. In this simple example, the first decision tree suggests that when Blueprint, Item A is Cardiovascular and the Jensen-Shannon Distance is low, then the two items are enemies. By contrast, the third decision tree suggests that all Cardiovascular-Infectious Diseases item pairs are enemies.

After resampling many times and growing decision trees in this manner, the resulting forest of decision trees is used to classify item pairs; if the majority of decision trees indicate that a given pair of items are enemies, then they are classified as enemies.

There is currently no best practice for identifying enemy items. It is the intention of this study to determine if how well a combination of LDA and a random forest model will perform at identifying enemy item pairs. To that end, the following chapter will describe a methodology answering the following research questions:

A. Do random forest models adequately identify and classify enemy item pairs?

B. Do metrics derived from Latent Dirichlet Allocation (namely, Jensen-Shannon Distance) improve a random forest models ability to classify enemy item pairs?

C. Can random forest models be retrained, using subject matter input, to improve their ability to classify enemy item pairs.

CHAPTER III

METHODS

The following chapter describes the methods that will be employed in this study. Those methods can be classified into the following categories: 1) data collection and data cleaning, 2) the fitting of latent Dirichlet allocation (LDA) and random forest models, 3) initial classification of item pairs, 4) subject matter expert (SME) assessment of item pair classification, and 5) random forest model retraining and item pair re-classification.

The National Commission on Certification of Physician Assistants (NCCPA) will provide the dataset used in this study, as well as access to the subject matter experts (SMEs) who will provide feedback on the results. It is with the expressed permission of NCCPA that their name and logo is used in the following chapter.

Cleaning and Preparing the Data

A sample of 7,205 items, drawn from a national examination item bank, have been identified for this study. Lai & Becker (2010) found that small samples of items limited their ability to sufficiently classify enemy item pairs, so the present study places an emphasis on utilizing a larger sample. It is expected that a larger sample will provide a reasonable reflection of the myriad latent topics that might be found in an operational item bank.

Furthermore, while the sample drawn for this study will not strictly adhere to the proportions prescribed in the exam blueprints (since the item bank itself does not adhere

exactly to these proportions), we should expect that the sample will roughly align with the blueprint specifications. If, for instance, the blueprint specifies that nearly 20% of the items on a given form should focus on "Formulating Most Likely Diagnosis," and the sample of items is comprised of only 2% that do so, more of those items will be drawn from the bank to bring the proportion more in line with expectations. The two dimensions of the blueprint for the items bank in question are found below in Tables 3 and 4.

Table 3. Organ System Blueprint Specifications

Organ System	% of Exam	% of Sample	Sample N
Cardiovascular	16%	14%	990
Dermatologic	5%	8%	545
EENT (Eyes, Ears, Nose and Throat)	9%	6%	419
Endocrine	6%	7%	491
Gastrointestinal/Nutritional	10%	12%	833
Genitourinary	6%	6%	417
Hematologic	3%	7%	494
Infectious Diseases	3%	11%	795
Musculoskeletal	10%	8%	607
Neurologic System	6%	4%	258
Psychiatry/Behavioral	6%	7%	511
Pulmonary	12%	7%	514
Reproductive	8%	4%	300
Other	<1%	<1%	31
Total	**100%**	**100%**	**7,205**

Table 4. Task Blueprint Specifications

Task	% of Exam	% of Sample	Sample N
History Taking & Performing Physical Examinations	16%	23%	1,672
Using Laboratory & Diagnostic Studies	14%	16%	1,136
Formulating Most Likely Diagnosis	18%	20%	1,474
Health Maintenance	10%	5%	363
Clinical Intervention	14%	9%	642
Pharmaceutical Therapeutics	18%	17%	1,206
Applying Basic Science Concepts	10%	10%	685
Legal\Ethical	< 1%	<1%	27
Total	**100%**	**100%**	**7,205**

For each item, the item stem string and the item answer string will be concatenated into a single string, constituting a single, albeit brief, document. In addition to the stem and answer strings, the following metadata fields will be retained: 1) item ID, 2) a list of item IDs that SMEs have determined are enemies of a given item, 3) the diagnosis on which the item focuses, 4) the International Classification of Diseases and Related Health Problems, Ninth Revision (ICD-9) code with which the item is associated, 5) Organ System and 6) Task, the two blueprint dimensions with which items are affiliated, 7) the author of the item, and 8) the source from which the item was developed, where available. These same classifications are used in operational test development and assembly for this national examination.

Terms within stem-answer strings will be stemmed using a Porter stemmer (Porter, 1980). This process reduces terms to a common root word. To the LDA

algorithm, walk, walk***ed***, walk***ing***, walk***er,*** and walk***s*** are five distinct terms. After stemming, all five terms are reduced to "walk", and the LDA is more able to identify relationships between documents in which these terms appear. The Porter stemmer (Porter, 1980), as implemented in the SnowballC (Bouchet-Valet, 2014) package for R, will be used to stem all terms in the text corpus, which is defined as all text data found in the stems and correct answers found in the 7,205-item sample.

Stem-answer strings will be pruned of stop words—frequent terms, such as "and," "are," "see," and "which"—using a combination of commonly used lists of stop words. The Stop Word List 1 from the Onix text retrieval toolkit (404 words), the SMART information retrieval system (571 words), and the Snowball compiler (174 words) are used to strip common stop words from items. In addition, after briefly assessing term frequencies, a custom stop word list will be generated based on the specific context of the item bank. That is, because of the specialized nature of the exam in question, certain words and numerals are used so frequently, or used so infrequently, across item strings as to become essentially meaningless. For example, if the term *mandibular* is used in only one item, it doesn't contribute to the topic model; in the same way, if the term *patient* occurs in every item, it also will not contribute to the topic model.

As a final cleaning step, the data set is transformed into a document-term matrix (DTM), with a row for each item (document), and a column for each term in the text corpus; at the intersection of a given document and term, an integer value indicates the number of times the term has been assigned to that document. This DTM is sparse, as most documents contain relatively few terms from the overall text corpus. This is

especially true for very short documents, such as test items. Intuitively, row sums provide a word count for each document, and column sums provide the frequency of use for each term in the text corpus.

Latent Dirichlet Allocation

After raw item data are cleaned, the latent Dirichlet allocation (LDA) model will be fit to the DTM (Silge & Robinson, 2017). When fitting the LDA model, variational expectation maximization (VEM) will be used with a burn in of 1,000 iterations, and an additional 1,000 iterations thereafter. An empirically determined *alpha* prior equivalent to the inverse of the number of topics $\left(\alpha_{prior} = \frac{1}{k}\right)$ is used. This provides a starting point where the assumption is that only one topic is likely to contribute to any given item; assuming something approaching item writing best practices, this is a reasonable starting point. A consistent seed is provided in order to facilitate replication of results. To expedite the estimation, parallel processing is enabled by splitting computational threads among all available CPU cores.

To build the LDA model, the number of latent topics must be specified beforehand. While there is no one best way to determine the ideal number of latent topics, three indices (Arun et al., 2010; Brown, Della Pietra, Della Pietra, Lai, & Mercer, 1992; Cao et al., 2009) will be used to assess the fit of several models across a range of specified topics.

All three fit indices will be assessed simultaneously and repeatedly using a *k*-fold cross validation approach in the following manner. The document-term matrix will be randomly partitioned into five equal folds. For each specified number of topics, the LDA

will be fit using four of the five folds. The model will then be applied to the holdout fold, constituting 20% of the documents, and the fit indices will be applied to these results. This process will repeat five times, with each fold serving one time as the holdout set. Where the means of the indices are minimized, the number of topics is likely to represent a range of values that may be used to fit the LDA model using the entire dataset.

Having settled on a range of reasonable topic numbers, performance of the model will be assessed by observing the distribution of γ estimates (where γ is the probability that an *item* was generated by a given latent topic). The LDA model, by definition, generates a document-by-topic matrix, where each row represents a document, and the probability that the document was generated by each of the specified latent topics. For instance, for 50 specified topics, the model returns 50 γ estimates for each item, effectively creating a semantic fingerprint in the form of a probability distribution that can be compared to other fingerprints, along with available metadata, to flag items that express a high degree of similarity. Figure 2, on the following page, provides an example of γ distributions for 25 documents with five topics. Each panel represents a document, and within each panel, the y-axis, ranging from 0 to 1, presents the probability that each of the five topics, shown along the x-axis, contributed to the document. Documents 17 and 22, highlighted in the table, are clearly very different: Topic 3 is, far and away, the most likely contributor to Document 17, while that same topic likely contributes very little, if at all, to Document 22. By contrast, documents 24 and 25, highlighted in the table, are very similar, with Topic 5 driving the word contributions to the document.

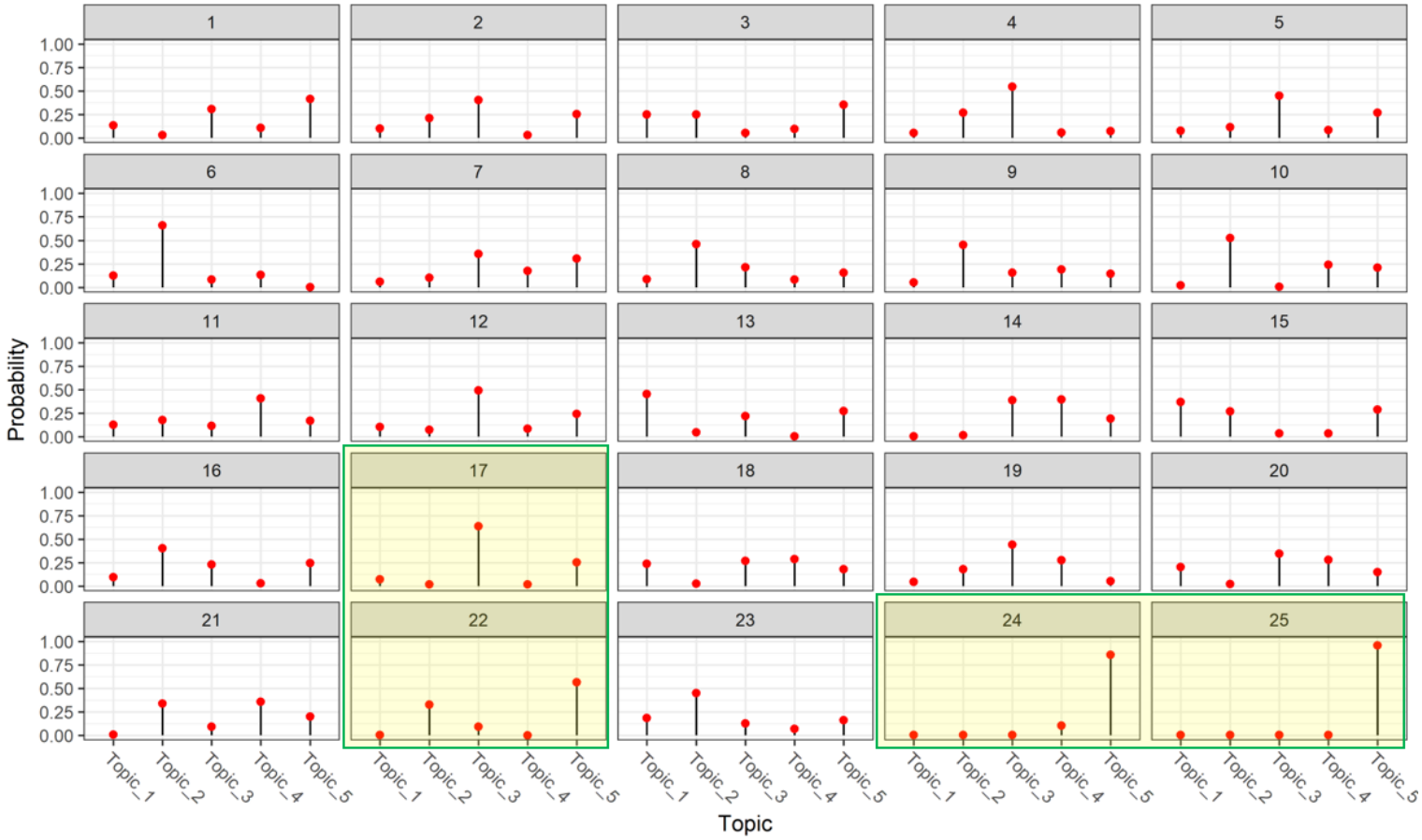


Figure 2. Example Gamma Estimates

Having estimated γ distributions for each item, the proximity of one distribution to another is assessed by calculating the Jensen-Shannon distance (JSD), (the square root of the Jensen-Shannon divergence). For the sample of 7,502 items examined in this study, 51,912,025 pairwise distances will be calculated. These JSD values provide a final check of the LDA model. If two items are calculated to be in close proximity to one another, we would expect that the items and their correct answer strings, what we defined as the item document in the data cleaning step, should be similar to one another. A visual inspection of close items will be undertaken to verify that the model is drawing reasonable conclusions about the topics that contribute to nearby items. If those conclusions are not reasonable, the LDA will be refit using another value for the number of topics, as determined by the analyses of fit, above.

For this study, item distances at or below 0.2 are flagged as possible enemies. The veracity of the model is tested by assessing the sensitivity of the model: a measure of whether the items flagged by the model capture a reasonable proportion of those item pairs that were classified as enemies by subject matter experts. Furthermore, since subject matter experts have not classified all enemy pairs, in a later step we will assess the degree to which the model reasonably flags item pairs that subject matter experts *would* flag.

Random Forests

Having considered the degree to which the Jensen-Shannon distance accurately identifies enemy item pairs, the study turns to the second research question: Does this measure of lexical distance, as generated by a latent Dirichlet allocation model, allow a

data mining approach to predict enemy item pairs more accurately? To this end, the 51,912,025-row, item pair-by-distance matrix is merged with item metadata in order to train a random forest model. These data are:

1) a categorical variable that indicates the diagnosis on which the item focuses (2,758 unique diagnoses), from common diagnoses, such as "asthma," and "pneumonia," to less common diagnoses, such as "abscess, perirectal";
2) a numeric variable that indicates the International Classification of Diseases and Related Health Problems, Ninth Revision (ICD-9) code with which the item is associated (1,933 unique ICD-9 designations);
3) a categorical variable indicating the first dimension of the blueprint variables, Organ System (14 unique systems, see Table X);
4) a categorical variable indicating the second of two dimensions of the blueprint variables, Task (8 unique tasks, see Table X);
5) a categorical variable indicating the author of the item (129 unique authors, including the designation "Historical Data" which accounts for roughly 16% of all items);
6) and the source from which the item was developed (4,084 unique references), where available; roughly 38% of items have no associated reference.

In addition to the JSD, for all item pairs there are two columns for each of the above variables (e.g. ICD9_A & ICD9_B). Except for JSD and ICD9, all of the above variables are treated as categorical. JSD is naturally a continuous variable, and ICD9 is also treated as continuous, as similar items tend to have different, though numerically

close, values. That is, when training the random forest model, if the decision trees are allowed to read the ICD9 code as continuous, items that have different ICD-9 codes (555.0 – Crohn's disease, small intestine and 555.1 – Crohn's disease, large intestine), are more likely to be treated as similar.

In preparation for the data mining step, a sample that will be used for training and validation is created from the larger sample of all item pairs. This step is taken because in the dataset containing *all* item pairs, the target variable that will be used to train the model (a dichotomous indicator of whether or not subject matter experts have classified the pair as enemies) is positive in roughly 0.01% of all cases. In order to train the model with a more useful proportion of positive target values, a new sample will be created wherein all enemy items constitute approximately 10% of the dataset, and a random selection of items that have not been flagged as enemies will constitute the remaining 90%.

The data are then partitioned, 55% for training a random forest model, and 45% for validating the model. Two random forest models will then be trained, one with the Jensen-Shannon distance, and one without. All other model parameters and hyperparameters are identical. Having trained and validated the models, the full data set containing all item pairs is scored using each model.

Models are compared to one another through metrics of classification accuracy and error. Receiver operator characteristic (ROC) indices—e.g. the area under a ROC curve will provide an indication of which model performs better. In addition, confusion matrices will provide insight into sensitivity, specificity, and accuracy of the models.

The overwhelming majority item pairs, we assume, are not enemies; by extension, accuracy and specificity are likely to be high, regardless of the model, because True Negative rates will be high. On the other hand, sensitivity, the true positive rate, will be an especially good indicator of misfit, since enemies are rare.

Subject Matter Experts and Truth

When we train the random forest model, we only know which item pairs are enemies, as classified by subject matter experts (SMEs). For the remaining item pairs, which constitute roughly 90% of the training and validation data, we do not know which pairs are in fact not enemies, and which ones are enemies but have never been assessed by SMEs. That is to say, because item pairs that have been assessed by SMEs are only labeled as enemies, and unlabeled if they are not, in training the random forest model with pairs about which we do not know truth, we are potentially training it incorrectly. We are very likely telling it that some pairs are not enemies, when in fact they are. As a result, we should expect that the model is imprecise.

After training the random forest model, it will then be applied to all item pairs to classify them as enemies, or not enemies. Because we expect the model to be imprecise, in cases where the model indicates that item pairs are highly likely to be enemies, (i.e., a likelihood of .9 to 1), but those items have not been classified as enemies by SMEs, those items will be presented to SMEs who have experience writing items. The sample of item pairs will be reduced further to exclude non-unique item pairs. That is, when an item is paired with itself, we expect a high likelihood of enemy classification, and we don't need SMEs to review this classification. In addition, the model does not make a distinction

between Item 12 paired with Item 24, and Item 24 paired with Item 12. While all permutations or enemy pairs will be assessed by the random forest model, only unique combinations will be assessed by SMEs.

Twenty SMEs have agreed to assess item pairs over the course of one week. In addition to being experts in the content found in the sample items, all SMEs have served on exam development committees, including item writing, key validation, and forms review committees. The SMEs, therefore, have not only a keen sense of the technical material found in the items, but they are also familiar with the questions this study is trying to address.

An algorithm, similar to an automated test assembly engine, will provide enemy pairs to SMEs, favoring item pairs that have been assigned a high likelihood of being enemy pairs by the random forest model. Where item pairs have the same likelihood, the engine favors item pairs that have a lower Jensen-Shannon distance. In this manner, all item pairs that the models suggest are more likely to be enemies are assessed first.

Each item pair will be assessed by at least two SMEs. In cases where the first two SMEs disagree about enemy classification, the engine will serve the pair to a third SME for adjudication. SMEs have agreed to work for eight hours providing feedback on these item pairs. In return for their efforts, each SME will be offered a $500 honorarium.

SME input would be financially impossible if travel and lodging were required. Therefore, an app was developed that will allow SMEs to login remotely using secure credentials. These credentials not only serve to protect the operational items, but they also allow the algorithm to differentiate between SMEs so that it can serve unique item

pairs to any given SME only one time. An example of the interface is shown in Figure 3, below.

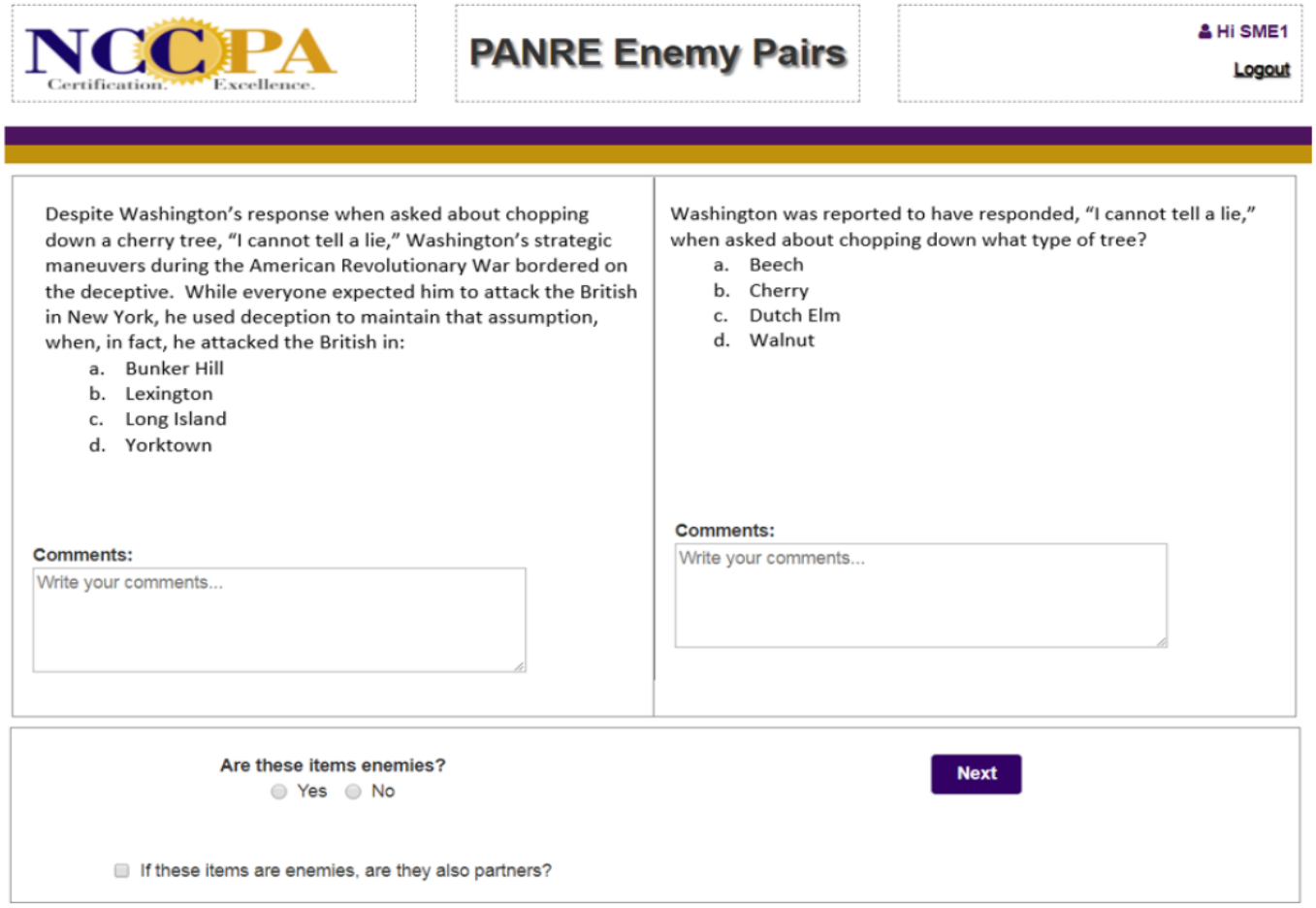


Figure 3. Enemy Item Classification Application

The interface prompts SMEs for feedback on item pairs in terms of enemies; in addition, it asks for feedback on item partnership. By definition, item partners are also enemies, but they have the additional characteristic of being ideal items for use in remediation. That is, if an examinee answers an item incorrectly, a partner item might be a good candidate for retesting the content later. Item partnership is not a focus of the present study, but it is in the interest of the organization supporting this research to capitalize on the opportunity to gather this information from SMEs.

The identities of the SMEs, while known to the National Commission on Certification of Physician Assistants in order to provide honoraria, are not know to the principal investigator.

Retraining the Random Forest Model

After gathering information from SMEs regarding the classification of item pairs as enemies or non-enemies, the Random Forest model will be retrained. The newly confirmed enemy items will be added to the pool of items that have already been identified as enemies by SMEs. As before, a sample of item pairs will be assembled in order to train and validate the model. That sample will again be comprised of 10% enemy pairs, and 90% item pairs that have not been classified as enemies. Included in that 90% will be all item pairs that have been explicitly classified by SMEs as non-enemies, thereby capitalizing on all items about which we know truth.

After training the new model, the entire sample of item pairs will be reclassified. The new classification results will be compared to the classification of the previous random forest model, as well as the item pairs that are classified using the latent Dirichlet

allocation model and the Jensen-Shannon distance. Models will again be compared using metrics of classification accuracy and error. A confusion matrix will provide insight into sensitivity, specificity, and accuracy, and receiver operator characteristic (ROC) indices—e.g., the area under a ROC curve—provide an indication of which model performs better.

It is the intent of the methodology described in this chapter to adequately determine if metrics derived from Latent Dirichlet Allocation provide appreciable additional information to help classify enemy pairs.

CHAPTER IV

RESULTS

The following chapter describes the step-by-step results found in the process of 1) data collection and data cleaning, 2) the fitting of latent Dirichlet allocation (LDA) and random forest models, 3) initial classification of item pairs, 4) subject matter expert (SME) assessment of item pair classification, and 5) random forest model retraining and item pair re-classification.

Cleaning and Preparing the Data

A sample of 7,205 items was drawn from a national examination item bank. While the sample did not strictly adhere to the proportions prescribed in the exam blueprints (since the item bank itself does not adhere exactly to these proportions), Tables 5 and 6 show that the sample roughly aligns with the blueprint specifications, as we would expect of an operational item bank.

All items for this particular examination are coded along two blueprint dimensions; they are organ system (an umbrella category not entirely composed of organ systems), and task. Table 5, a breakdown of sample items across organ system categories, shows a loose distributional adherence to the blueprint, with four categories (Hematologic, Infectious Diseases, Pulmonary, and Reproductive) showing a difference of more than three percentage points from the target.

Table 5. Organ System Blueprint Specifications

Organ System	% of Exam	% of Sample	Sample N
Cardiovascular	16%	14%	990
Dermatologic	5%	8%	545
EENT (Eyes, Ears, Nose and Throat)	9%	6%	419
Endocrine	6%	7%	491
Gastrointestinal/Nutritional	10%	12%	833
Genitourinary	6%	6%	417
Hematologic	3%	7%	494
Infectious Diseases	3%	11%	795
Musculoskeletal	10%	8%	607
Neurologic System	6%	4%	258
Psychiatry/Behavioral	6%	7%	511
Pulmonary	12%	7%	514
Reproductive	8%	4%	300
Other	<1%	<1%	31
Total	**100%**	**100%**	**7,205**

Table 5, a breakdown of sample items across task categories, also shows a loose distributional adherence to the blueprint, with three categories (History Taking & Performing Physical Examinations, Health Maintenance, and Clinical Intervention) showing a difference of more than three percentage points from the target.

The sample item data set was read from an XLSX file into the R programming environment, and the following variables were retained: 1) item ID, 2) the item stem string, 3) the correct answer string, 4) a list of item IDs that SMEs have determined are enemies of a given item, 5) the diagnosis on which the item focuses, 6) the International Classification of Diseases and Related Health Problems, Ninth Revision (ICD-9) code with which the item is associated, 7) Organ System and 8) Task, the two blueprint

dimensions with which items are affiliated, 9) the author of the item, and 10) the source from which the item was developed, where available.

Table 6. Task Blueprint Specifications

Task	% of Exam	% of Sample	Sample N
History Taking & Performing Physical Examinations	16%	23%	1,672
Using Laboratory & Diagnostic Studies	14%	16%	1,136
Formulating Most Likely Diagnosis	18%	20%	1,474
Health Maintenance	10%	5%	363
Clinical Intervention	14%	9%	642
Pharmaceutical Therapeutics	18%	17%	1,206
Applying Basic Science Concepts	10%	10%	685
Legal\Ethical	< 1%	<1%	27
Total	**100%**	**100%**	**7,205**

The item stem string and correct answer string were concatenated to create a single text document per item. Table 7 shows the number of unique values found within each variable.

Table 7. Summary of Data

Variable	Unique Values
Item ID	7,205
Question + Answer String Documents	7,205
Items with Enemies	3,714
Diagnosis	2,757
ICD9	1,730
Organ System	14
Task	8
Author	129
References	4,084

The 7,205 items that comprise the sample are reflected in the totals for item IDs and the concatenated question + answer string documents. Approximately half of the items in the sample (3,714) were members of at least one enemy relationship; as enemy relationships are reciprocal, half that total (1,857) constitutes the total number of enemy relationships that were identified by subject matter experts prior to this study.

In all, 129 unique authors, using 4,084 unique references, wrote items addressing 2,757 unique diagnoses, 1,757 unique ICD9 classifications, and covering all 14 organ system and 8 task blueprint categories. It is this diversity and the information-dense quality of these variables that were used to train the models in this study.

After concatenating each stem string with the corresponding correct answer string, the resulting text documents were cleaned of any html encoding. Specifically, left- and right-hand angle brackets, and the text found between them (e.g., "<br>") were removed from all text documents. Similarly, strings enclosed by an ampersand and a semicolon (e.g.. " ") were removed. In all, 1,976 items (26% of the sample) were cleaned of html encoding in this manner.

The documents were then tokenized, a transformation of the dataset from a wide format to a long, one-document and one-token (word) per-row, format. The strings found in the resulting vector were stemmed using the Porter stemming algorithm (Porter, 1980), as implemented in the SnowballC (Bouchet-Valet, 2014) package for R. After stemming, the number of unique words in the dataset was reduced from 11,684 to 8,633 (Table 9). Overall word frequency—the number of times a word appears in the dataset, across all documents—increased from a mean of roughly 35 to 47. This result is unsurprising, as

words were effectively collapsed into their roots ("runs" and "runner" both became "run"), leaving the same total number of words as found in the raw data set, but fewer unique words.

The resulting vector of words was compared to a vector of stop words (those words that do not contribute substantively to the meaning of a text document); words common to both the sample and the stop word lists were removed from the sample. Stop word lists from three sources (Table 8) were used: the Onix Text Retrieval Toolkit ("Onix text retrieval toolkit: Stopword list 1," n.d.), the SMART information retrieval system (Buckley, 1985), and the Snowball compiler (Porter, n.d.). Between these three stop word lists there were a total of 728 unique stop words.

Table 8. Stop Word Sources

Lexicon	**N words**
Onix Stop Word List 1	404
SMART	571
Snowball	174
Custom Stop Words	2,423
Total unique	**3,151**

A custom stop word list was generated by identifying words with very low term frequency values. Stop words found in the above lists were eliminated from the custom stop word list, which in this final form consisted of very low frequency words and numerals, typically appearing in only one stem-answer string.

Across all stop-word lists, there were 3,151 unique terms, including strings consisting entirely of numerals. All four stop-word lists were combined to create a

master list which was used to filter stop words from the study sample. The removal of these terms reduced the vector of unique terms from 8,633 to 5,841 (Table 9).

Table 9. Stemming and Stop Word Removal Results

	Raw	Post Stemming (Step 1)	Post Stop Word Removal (Step 2)
Documents (Items)	7,205	7,205	7,205
Total number of words	416,006	416,006	227,432
Unique words	11,684	8,633	5,841
Mean word frequency	35.28	47.15	38.94
Min word frequency	1	1	1
Max word frequency	7,203	7,203	4,125
Mean word count per item	75	75	35
Min word count per item	8	8	1
Max word count per item	369	369	219
DTM Size (dimensions)	7,205 × 11,684	7,205 × 8,633	7,205 × 5,841
DTM Size (cells)	84,183,220	62,200,765	42,084,405
DTM % Reduction	0%	74%	50%

As a final cleaning step, the data set was transformed into a document-by-term matrix (DTM), with a row for each item (document), and a column for each term in the text corpus; at the intersection of a given document and term, an integer value indicates the number of times the term has been assigned to that document (Table 9). The DTM for the raw data set is composed of 7,205 rows (one per item/document), and 11,684 columns (one per unique word); each of the 84,183,220 cells contains an integer value representing the number of times a given word is used a given document (term frequency). After stemming and stop word removal, the DTM was reduced to 7,205 rows by 5,841 columns (42,084,405 values of term frequency). Stemming and stop word removal account for a 49.99% reduction of the DTM, reducing the computational load of the following latent Dirichlet allocation steps.

Latent Dirichlet Allocation

The following section describes the steps that were taken to fit a latent Dirichlet allocation (LDA) model to the sample data, from using fit statistics to determine a reasonable number of topics to specify for the model, to calculating the Jensen-Shannon distances for all pairs of items in the sample.

Determining the Number of Topics

To build the LDA model, the number of latent topics must be specified beforehand. To make this determination, the following process was used, and the ensuing results were found.

Three fit indices (Arun et al., 2010; Brown et al., 1992; Cao et al., 2009) were used to assess the reasonableness of eight models with topic (*k*) specifications ranging from ten to eighty (Table 10).

All three fit indices were assessed simultaneously and repeatedly using a *k*-fold cross validation approach in the following manner. The DTM was randomly partitioned at the document level into five sets. Four sets were used to train the model, and a holdout set (20% of the sample) was used to validate the model. To train the model, variational expectation maximization (VEM) was used (Appendix A), with a burn in of 1,000 iterations, and an additional 1,000 iterations thereafter. An *alpha* prior equivalent to the inverse of the number of topics $\left(\alpha_{prior} = \frac{1}{50}\right)$ was set. This process was repeated five times per model, with each fold serving four times as part of a larger training set, and once as a validation, or holdout, set. Across all models, this process was repeated a total of 40 times, with fit indices calculated for holdout sets each time.

The fit calculations were standardized by index so that they might be compared to one another on the same scale. The standardized results are found in Table 11; for each fit index, values are presented as calculated for the holdout sets for each model, with the means and standard deviations for each presented in italics. The means and standard deviations across all indices are presented at the bottom of the table in bold.

For all three fit indices, low values indicate better fit. Low mean values, therefore, suggest that across all folds a given model is performing better, and lower standard deviation values indicate more consistency across all validation sets for a given model.

According to the perplexity fit statistic, the model with 20 topics appears to fit best (mean = -0.36; sd = 0.43), whereas the Arun et. al. (2016) statistic indicates that a model with at least 80 topics is a better contender (mean = -1.06; sd = 0.08). Finally, the Cao Juan (2004) statistic suggests that 40 topics is ideal (mean = -0.63; sd = 0.14).

Table 10. Results of Topic Analysis

		Topics							
Index	**Fold**	**10**	**20**	**30**	**40**	**50**	**60**	**70**	**80**
Standardized Perplexity by Validation Fold	1.00	0.95	-0.86	-1.12	-1.04	0.67	1.01	1.14	1.83
	2.00	0.49	-0.33	-1.20	-2.37	-0.85	-0.74	-0.40	0.31
	3.00	0.96	-0.35	-0.77	-1.43	-1.00	0.54	0.87	1.08
	4.00	1.65	0.30	-0.15	-1.12	-0.26	0.71	1.03	1.61
	5.00	0.52	-0.56	-1.19	-1.35	-0.10	0.20	0.55	0.78
	Mean	*0.91*	*-0.36*	*-0.89*	*-1.46*	*-0.31*	*0.35*	*0.64*	*1.12*
	SD	*0.47*	*0.43*	*0.45*	*0.53*	*0.66*	*0.67*	*0.62*	*0.62*
Standardized Arun (2016) by Validation Fold	1.00	2.29	0.91	0.21	-0.12	-0.30	-0.56	-0.82	-1.01
	2.00	2.02	0.80	0.04	-0.14	-0.32	-0.67	-0.86	-1.07
	3.00	2.19	0.98	0.24	-0.06	-0.43	-0.54	-0.65	-0.97
	4.00	2.12	0.82	0.13	-0.27	-0.47	-0.66	-0.88	-1.11
	5.00	2.06	0.72	0.13	-0.37	-0.47	-0.79	-0.95	-1.17
	Mean	*2.14*	*0.84*	*0.15*	*-0.19*	*-0.40*	*-0.64*	*-0.83*	*-1.06*
	SD	*0.11*	*0.10*	*0.08*	*0.13*	*0.08*	*0.10*	*0.11*	*0.08*
Standardized Cao Juan (2004) by Validation Fold	1.00	3.00	0.49	-0.34	-0.46	-0.27	-0.41	-0.61	-0.56
	2.00	2.46	0.50	-0.37	-0.54	-0.26	-0.64	-0.65	-0.57
	3.00	2.14	0.49	-0.32	-0.61	-0.76	-0.29	-0.44	-0.64
	4.00	2.29	0.34	-0.30	-0.76	-0.49	-0.28	-0.51	-0.63
	5.00	2.34	0.32	-0.38	-0.77	-0.22	-0.33	-0.40	-0.57
	Mean	*2.45*	*0.43*	*-0.34*	*-0.63*	*-0.40*	*-0.39*	*-0.52*	*-0.59*
	SD	*0.33*	*0.09*	*0.03*	*0.14*	*0.23*	*0.15*	*0.11*	*0.04*
Mean		**1.83**	**0.30**	**-0.36**	**-0.76**	**-0.37**	**-0.23**	**-0.24**	**-0.18**
SD		**0.74**	**0.55**	**0.48**	**0.60**	**0.37**	**0.55**	**0.72**	**1.00**

These results, taken independently, provide enough justification to fit a final model with 20, 40, or 80 topics. Collectively, they offer a range of defensible topic specifications. In this case, the range was narrowed somewhat by calculating the mean across all fit statistics and all folds, shown in bold in the final rows of Table 10. The means of the three indices suggest an ideal topic specification of 30 to 50 topics.

The standard deviation of all fit statistic values was also calculated, and where this value is lower, the fit statistics as calculated across all training folds are less variant; that is, the statistics appear to converge in a manner that suggests some degree of consensus (Figure 4).

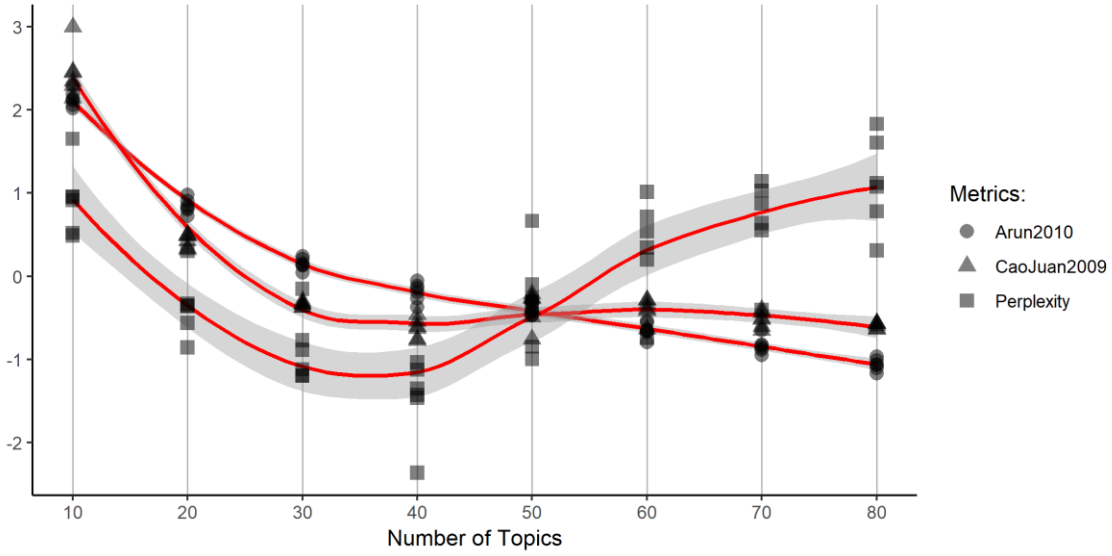


Figure 4. Results of Topic Analysis

Some ranges of topic specifications are clearly out of the question given the apparent lack of fit: a two-topic model and a one-thousand-topic model are patently inappropriate for the sample data. However, the well-known limitation of LDA is apparent here: there is no way to calculate a single best value to specify for an LDA topic model. Nevertheless, fit statistics, as presented in Table 11 and Figure 4, provide reasonable justification that 50 topics, the number of topics specified for the model in this study, is defensible.

Fitting the Final Model

Having settled on 50 topics, a final LDA model was fit. As in the prior step, a variational expectation maximization (VEM; Appendix A) approach was used to fit the model, with a burn in of 1,000 iterations, and an additional 1,000 iterations thereafter. An *alpha* prior equivalent to the inverse of the number of topics $\left(\alpha_{prior} = \frac{1}{50}, \text{or } 0.02\right)$ was set.

After fitting the model, the following results were found. The final estimate for *alpha* was 0.176; the product of *alpha* and the number of topics specified (*k*) provides an indication of how many topics typically contribute to a given item, in this case between eight and nine topics.

Beta estimates offer an indication of how related a term is to a topic; more specifically, the value of *beta* is the probability that a topic generated[3] a given term. The 5,841-word × 50-topic *beta* matrix (a sample of which may be found in Appendix B) provides that probability for each word and topic. One step toward assessing the reasonableness of the model is the review of the top *n* words from each topic. For instance, the top ten terms for Topic 46 are presented in Figure 5.

[3] Remember that latent Dirichlet allocation is a *generative* model; the items were generated, of course, by item writers, but the process is modeled as though the writers had 50 topic-buckets of words, and they wrote items by dipping into some combination of them.

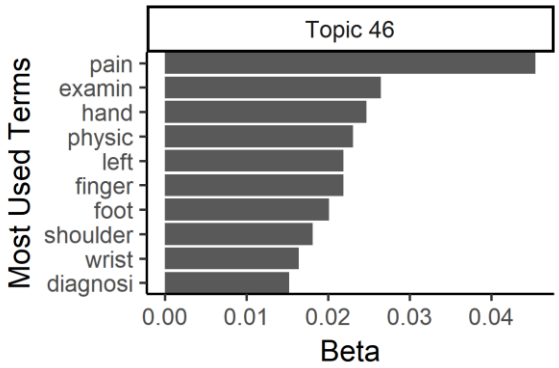


Figure 5. Top Ten Terms for Topic 46

This topic seems reasonable in that it appears to relate to pain in the extremities, a realistic topic about which, or around which, one might generate a question for a medical examination. A check of the top-*n* terms for all topics was undertaken, and all topics identified by the model appeared reasonable. The relevant plots may be found in Appendix C.

Gamma estimates offer an indication of how related a topic is to a document; more specifically, the value of *gamma* is the probability that a topic contributed to a given item. The 7,205-item × 50-item *gamma* matrix (a sample of which may be found in Appendix D) provides that probability for each item and topic. As is the case with *beta* estimates, one may assess the reasonableness of the model by reviewing the *gamma* estimates. For instance, the *gamma* values for Item 26 that are appreciably above zero are presented in Table 11. In effect, Topic 13 is responsible for 36% of the terms in this item, Topic 7 is the source of 33% of the terms, and Topics 9 and 44 together account for roughly 29%.

Table 11. Sample Topic Probability Distribution

	Topic 7	**Topic 9**	**Topic 13**	**Topic 44**
Gamma	0.334	0.113	0.360	0.180

If the topics highlighted by the *gamma* estimates for this item "hang together" in a rational manner, as informed by the *beta* estimates (see above), then this is more evidence in favor of the model.

Figure 6 presents the top-10 terms associated with Topics 7, 9, 13, and 44. Note that the x-axis scales are intentionally allowed to adjust by topic; the intent here is to show relative importance of a word within a topic, not importance across all topics. The words with the highest *beta* values for Topics 7, 13, and 14, which the model indicates are responsible for 87% of the words in the item, suggest symptoms related to cardiovascular distress. Topic 9, while not obviously related to the other topics, is not wildly out of place, either.

While it would have been impractical to cross reference the *gamma* estimates with the *beta* estimates for all 7,205 items in the sample, several were assessed in this manner to make certain that the model behaved reasonably.

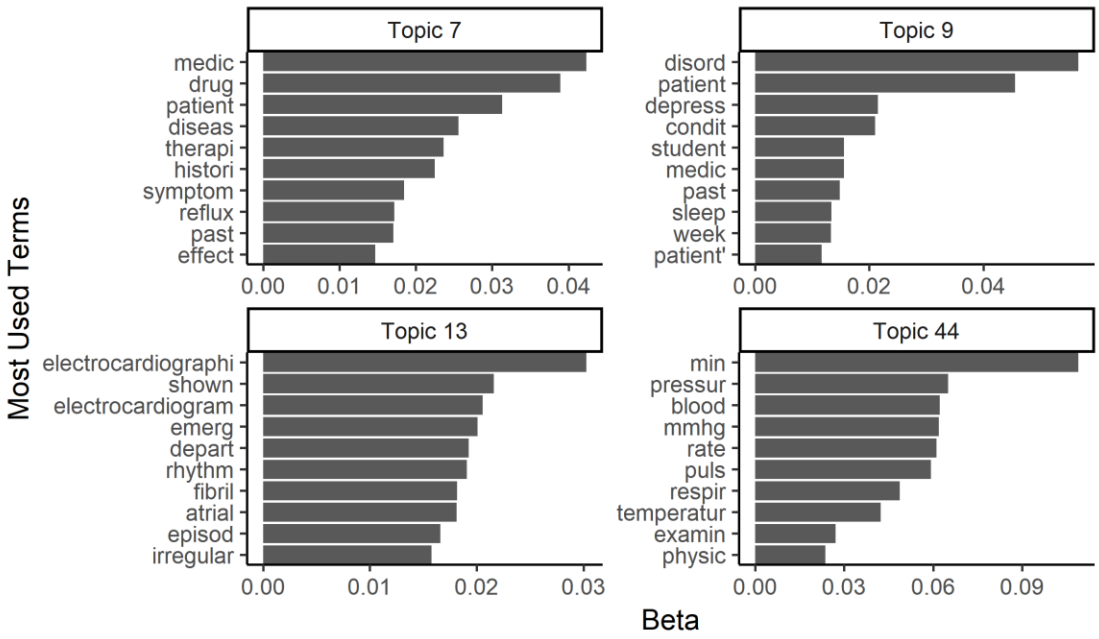


Figure 6. Top Ten Terms for Topics 7, 9, 13, and 14

Following the checks discussed above, the *gamma* estimates were used to calculate the Jensen-Shannon distances (JSD) for each pair of items. The mean JSD between all item pairs was 0.643 with a standard deviation of 0.092; the median was 0.643 (Table 12).

Table 12. Descriptive Statistics of Jensen-Shannon Distances

N-item pairs	JSD Mean	JSD Median	Standard Deviation	N-item pairs < 0.2	%-items < 0.2
51,912,025	0.606	0.643	0.092	316,711	0.610%

Descriptive statistics indicate that the distribution of JSDs is negatively skewed, as is apparent in a histogram of the distances (Figure 7). This distribution is expected. JSD is a measure of lexical similarity between documents; where JSD equals zero, two

documents are identical, and where JSD equals one, the two documents have nothing in common. It comes as no surprise, then, that the items in a test bank—all of which are written to the same blueprint targets—should bear some lexical similarity to one another while still exhibiting overall difference amongst most pairs.

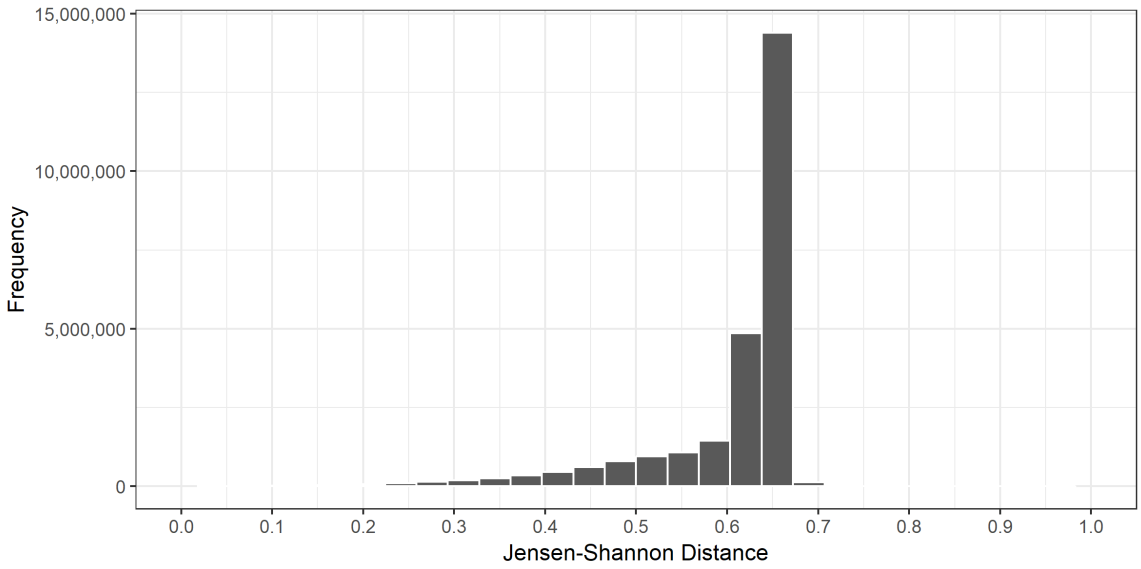


Figure 7. Histogram of Jensen-Shannon Distances Between All Combinations of Items

The purpose of this study is to identify enemy item pairs; ostensibly, most enemy items pairs are near one another. Approximately 0.6% of item pairs, (N = 316,711) have a JSD of 0.2 or below (Table 12). The distribution of the close items is not visible in the histogram of all item pairs (Figure 7), but a similar negatively skewed distribution is found when close items are presented in isolation (Figure 8).

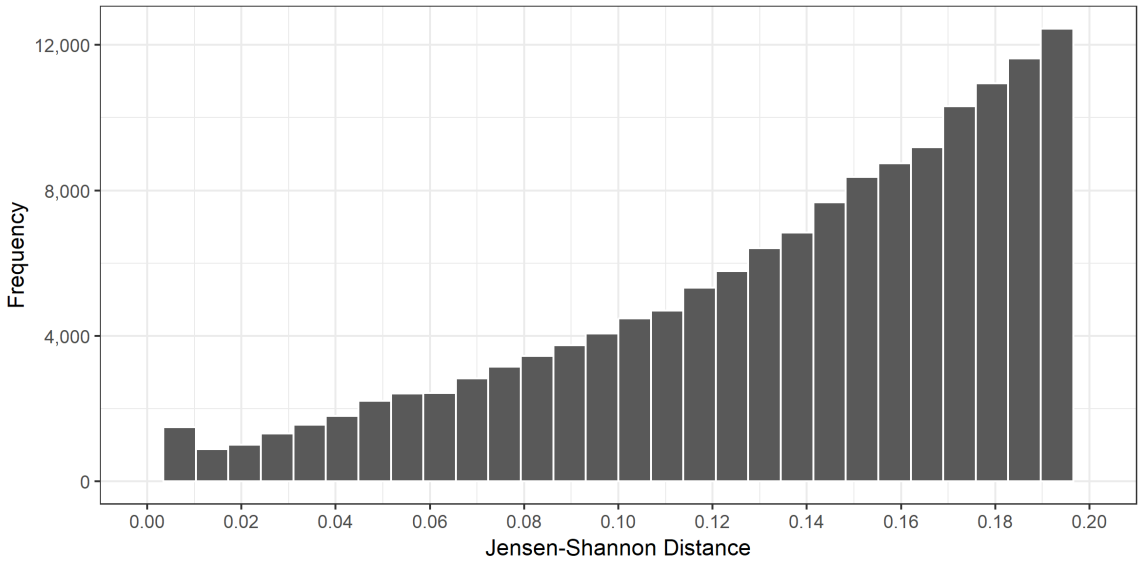


Figure 8. Histogram of Jensen-Shannon Distances Below 0.2

Fitting the First Round of Random Forest Models

Having fit the latent Dirichlet allocation model and having calculated all pairwise item Jensen-Shannon distances, the training of two random forest classifiers was undertaken. In order to train the models with a useful proportion of positive target values (item pairs that had been flagged as enemies by content experts), a sample of item pairs was created wherein all enemy pairs constituted approximately 10% of the dataset, and a random selection of items that were not flagged as enemies constituted the remaining 90%.

As described in the previous chapter, the following variables were used to train and validate the first classifier: Diagnosis, ICD-9, Organ System, Task, Author, Source, Enemy. For the second classifier, these variables were used, as well as the Jensen-Shannon distance.

The item pair data were partitioned; 55% of the sample was reserved for training a random forest model, and 45% to validate the model. Two random forest models were then trained. All other model parameters and hyperparameters were identical. Having trained and validated the models, the full data set containing all item pairs was scored using each model.

To assess the performance of the random forest classifier, the complete sample of approximately 52 million item pairs was reduced to roughly 26 million pairs by dropping duplicate pairs (e.g., Item 1 & Item 2 is a duplicate of Item 2 & Item 1) and pairs that consisted of a common item (e.g., Item 1 & Item 1).

From this reduced data set, confusion matrices were generated to assess the quality of the models. The first random forest classifier, trained without JSD values, correctly classified 99.88% of the enemy pairs (Table 13).

Table 13. Confusion Matrix, No JSD, Before SME Feedback

		Actual: No	Actual: Yes	Total
Predicted: No	**Frequency**	25,921,117.00	259.00	**25,920,000.00**
	%	99.88	0.00	**99.88**
	Row %	100.00	0.00	
	Column %	99.88	30.40	
Predicted: Yes	**Frequency**	30,441.00	593.00	**31,034.00**
	%	0.12	0.00	**0.12**
	Row %	98.09	1.91	
	Column %	0.12	69.60	
Total	**Frequency**	**25,951,558.00**	**852.00**	**25,952,410.00**
	%	**100.00**	**0.00**	**100.00**

This model incorrectly classified 259 pairs as non-enemies though content experts labeled them as enemies. In addition, while this model correctly identified 593 enemy pairs, it indicated that an additional 30,441 pairs were enemies. Sensitivity, specificity, and accuracy offer efficient ways to capture a classifier's performance. Sensitivity is the true positive rate $\left(\frac{TP}{TP+FP}\right)$, specificity is the true negative rate $\left(\frac{TN}{TN+FN}\right)$, and accuracy is the proportion of all correct classifications out of all classifications $\left(\frac{TP+TN}{TP+TN+FP+FN}\right)$. With so few enemies, specificity and accuracy should always be quite high. Sensitivity, by contrast, is the measure that will be most affected by an improvement in classification in this context. For the random forest classifier trained without the item pair JSD, sensitivity was 0.7, or 70% correct classification of enemy pairs (Table 14).

Table 14. Classifier Performance, No JSD, Before SME Feedback

Sensitivity	**Specificity**	**Accuracy**
0.70	1.00	1.00

A second random forest classifier was trained using the same training and validation sets as the first model with the one exception that the JSD values were included. The model correctly classified 99.89% of the enemy pairs (Table 15). The false negative rate improved considerably, as the model incorrectly classified 89 pairs that content experts labeled enemies, down from 259 pairs. In addition, while this model correctly identified 763 enemy pairs, it indicated that an additional 289,249 pairs were enemies.

Table 15. Confusion Matrix, With JSD, Before SME Feedback

		Actual: No	Actual: Yes	Total
Predicted: No	**Frequency**	25,662,309.00	89.00	**25,660,000.00**
	%	98.88	0.00	**98.88**
	Row %	100.00	0.00	**0.00**
	Column %	98.89	10.45	**0.00**
Predicted: Yes	**Frequency**	289,249.00	763.00	**290,012.00**
	%	1.11	0.00	**1.12**
	Row %	99.74	0.26	**0.00**
	Column %	1.11	89.55	**0.00**
Total	**Frequency**	**25,951,558.00**	**852.00**	**25,952,410.00**
	%	**100.00**	**0.00**	**100.00**

For the random forest classifier trained with the item pair JSD, sensitivity improved appreciably from 0.7 to 0.9, or 90% correct classification of enemy pairs (Table 16). Meanwhile, specificity and accuracy both suffered slightly, dropping to 0.99 each. Driving this change is a proportionally modest, though in practical terms quite large, increase in the overall number of items flagged as enemies. This figure rose from 0.12% of the sample to 1.12%, or from 31,034 to 290,012 flagged pairs.

Table 16. Classifier Performance, No JSD, Before SME Feedback

Sensitivity	Specificity	Accuracy
0.90	0.99	0.99

ROC curves provide a concise visual representation of the overall quality of a classifier. The random forest classifier provides a likelihood that a given pair of items are enemies; the ROC curve plots the relationship between the true positive rate (sensitivity)

and the false positive rate at 1% intervals for all item pairs. The improvement of the second model, trained with the JSD, over the first model is evident in Figure 9.

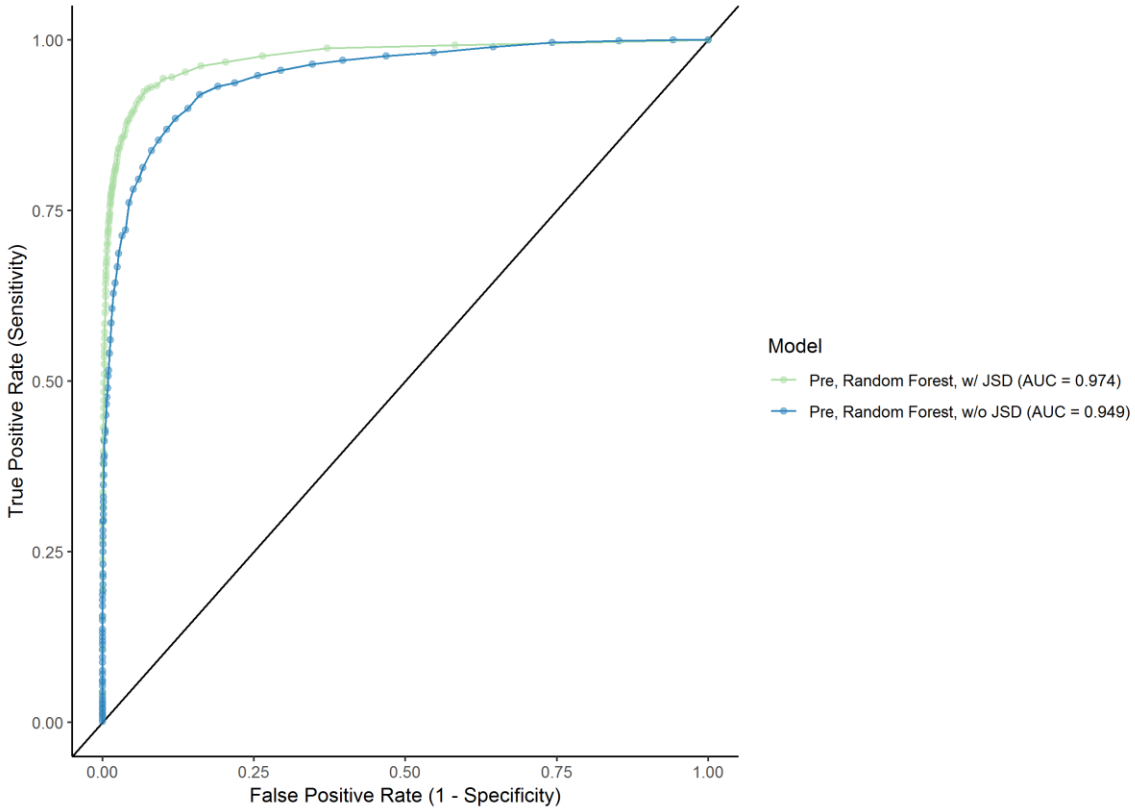


Figure 9. Empirical ROC Curves, Before SME Feedback

Feedback from Subject Matter Experts

As discussed in the previous chapter, one limitation of the way enemy pairs are recorded in the sample item bank is that for all pairs that were reviewed by subject matter experts (SMEs), only the enemy pairs were logged. That is, even if SMEs reviewed a pair of items and determined that they were not enemies, this decision was not recorded. Furthermore, because there are far more item pairs than can reasonably be assessed by SMEs, most enemy pairs have never been assessed. For these reasons, there is a surfeit

of item pairs with no indication that they have any enemy. The issue is that these items are used to train the random forest classifier as though they are not enemies. The degree to which this introduces error is unknown, and how best to address the issue?

A sample subset was created that included item pairs that 1) had the highest likelihood of being enemies ($p \geq 0.9$), 2) had a small Jensen-Shannon distance (JSD $\leq$ 0.2), and 3) were not already classified as enemies. To this subset was added item pairs that 1) had a lower likelihood of being enemies and 2) were previously flagged as enemies by SMEs. A total of 4,980 items were selected for review by SMEs.

These enemy pairs were then presented to SMEs[4] via a web-based application. Item pairs appeared side-by-side with correct answers highlighted. SMEs were asked to indicate if the pair of items were enemies. Item pairs were reviewed by a minimum of two SMEs; where there was disagreement, the item pair was served to a third reviewer for adjudication.

A total of 29 SMEs reviewed an average of 321 items apiece (Table 18). The distribution of items reviewed by SMEs was positively skewed, the result of two eager SMEs who reviewed 1,065 and 2,373 items, respectively (Appendix D). SMEs logged classifications a total of 9,309[5] times (Appendix D), contributing to the classification of 3,777 unique items out of 4,980 (75.8%). On average, SMEs classified item pairs as enemies 26.75% of the time (Table 17).

[4] By design, specific demographic information about the SMEs is unknown to the researcher. However, all SMEs were recruited from a pool of experts who have experience writing and/or reviewing items and forms that draw on the item bank from which the items in this study were sampled.
[5] This figure includes all classifications: a minimum of two responses per item pair, and in some cases a third response when adjudication was necessary.

Table 17. SME Feedback

N Raters	Mean Items	Median Items	SD Items	Mean % Enemies	Median % Enemies	SD % Enemies
29	321	194	443.21	26.74%	25.67%	14.37%

Of the item pairs that were classified, 761 (20.15%) were classified as enemies by SMEs (Table 18). Between enemy pairs and non-enemy pairs, there was no meaningful difference between the Jensen-Shanon distances, nor the random forest classifier probabilities of being enemies.

Table 18. Enemy Pairs, Random Forest Classification (*p*), and JSD

Enemies	N	%	Mean Probability	Mean JSD
No	3016	79.85%	93.40%	0.09
Yes	761	20.15%	93.59%	0.08

Figure 10 provides a graphical overview of the progress SMEs made through the sample of item pairs. Each dot represents an item pair, with the x-axis representing the Jensen-Shannon distance, and the y-axis representing the likelihood that an item pair were enemies, as estimated by the random forest classifier. Red dots indicate that the SMEs determined that the pair of items were not enemies, blue dots indicate that the items are enemies, and green dots indicate that the review process was not completed; these last items needed at least one more review by an SME. Note that the algorithm that served enemy pairs to SMEs chose high-likelihood items first; this is why incomplete item pairs systematically appear at the bottom of the plot.

Having sought the feedback of subject matter experts and incorporating their classifications into the dataset, new random forest classifiers were trained and validated. The data set used to train and validate the models was the same as the prior dataset, with the exception that enemy classifications were updated.

Again, the item pair data were partitioned; 55% of the sample was reserved for training a random forest model, and 45% to validate the model. Two random forest models were then trained. All other model parameters and hyperparameters were identical. Having trained and validated the models, the full data set containing all item pairs was scored using each model. Again, the complete sample of approximately 52 million item pairs was reduced to roughly 26 million pairs by dropping duplicate pairs and pairs that consisted of a common item. From this reduced data set, confusion matrices were generated to assess the quality of the models.

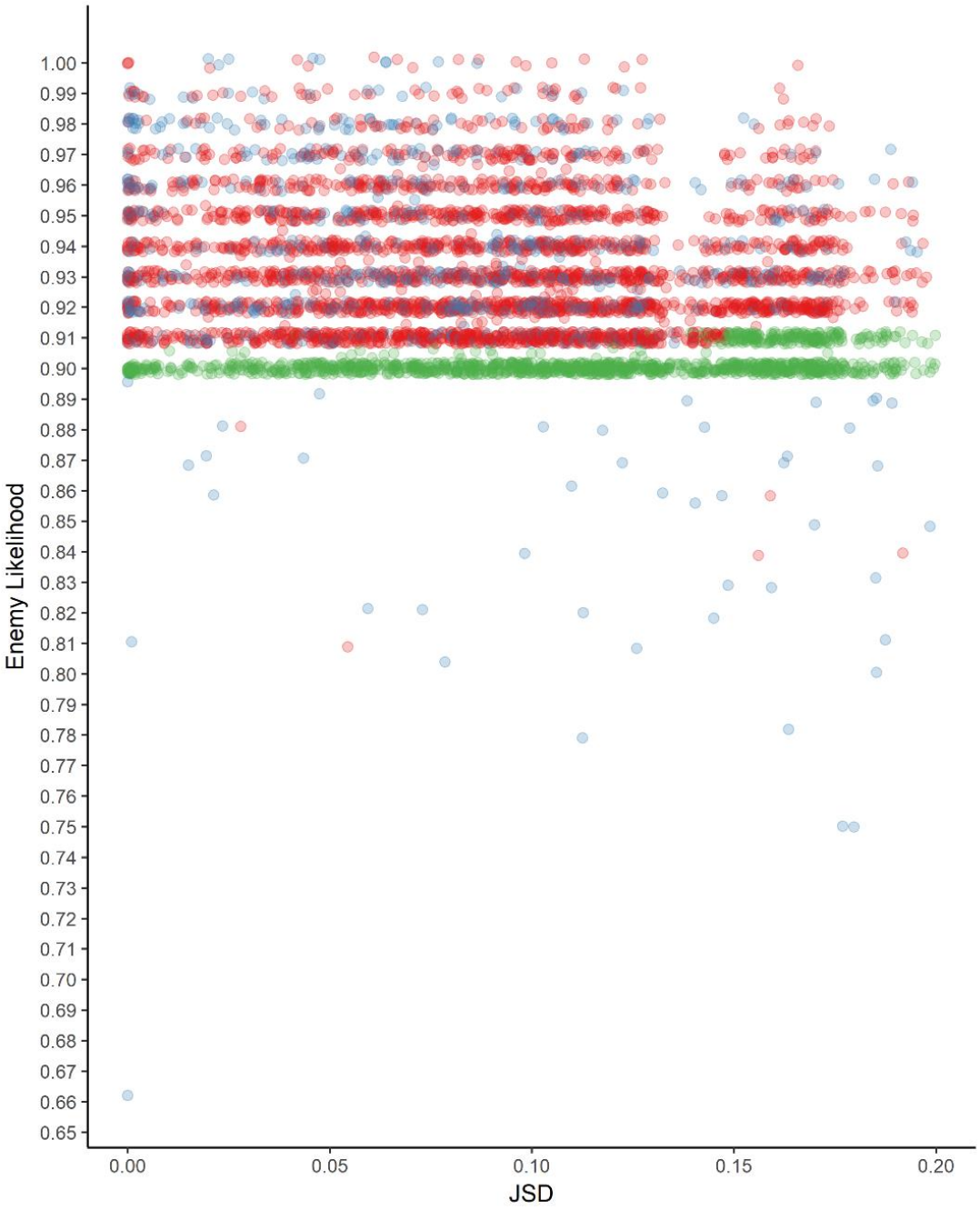


Figure 10. SME Ratings, Enemy Likelihood by JSD

Fitting the Second Round of Random Forest Models

The third random forest classifier, Model 3 (the first since incorporating new enemy pair data), trained without JSD values, but with SME input, correctly classified 99.82% of the enemy pairs (Table 19). The false negative rate worsened somewhat, as the model incorrectly classified 344 pairs (21.86%) that content experts labeled enemies, up from 10.45% of pairs in Model 2, which used the JSD, but no SME input, and somewhat better than Model 1 (30.4%) which used neither the JSD values, nor any SME input. While this model correctly identified 1,230 enemy pairs, it indicated that an additional 45,975 pairs might be enemies, down from 289,249 pairs in Model 2, though up from the 30,441 pairs of Model 1.

Table 19. Confusion Matrix, No JSD, After SME Feedback

		Actual: No	Actual: Yes	Total
Predicted: No	**Frequency**	25,904,861.00	344.00	**25,910,000.00**
	%	99.82	0.00	**99.82**
	Row %	100.00	0.00	
	Column %	99.82	21.86	
Predicted: Yes	**Frequency**	45,975.00	1,230.00	**47,205.00**
	%	0.18	0.00	**0.18**
	Row %	97.39	2.61	
	Column %	0.18	78.14	
Total	**Frequency**	**25,950,836.00**	**1,574.00**	**25,952,410.00**
	%	**99.99**	**0.01**	**100.00**

For the random forest classifier trained without the item pair JSD, but with SME input (Model 3), sensitivity worsened appreciably from Model 2's 0.9 to 0.78, or 78% correct classification of enemy pairs (Table 20). Meanwhile, specificity and accuracy

both improved slightly, increasing to 1.00 each. Driving this change is a decrease in the overall number of items flagged as enemies. This figure fell from 1.12% of the sample in Model 2 to 0.18%, or from 290,012 to 47,205 flagged pairs.

Table 20. Classifier Performance, No JSD, With SME Feedback

Sensitivity	Specificity	Accuracy
0.78	1.00	1.00

The fourth random forest classifier, Model 4, trained with JSD values and with SME input, correctly classified 99.72% of the enemy pairs (Table 21). The false negative rate improved somewhat, as the model incorrectly classified 170 pairs (10.80%) that content experts labeled enemies, an improvement over Model 3 (21.86%), and roughly equivalent to Model 2 (10.45%), which used the JSD, but no SME input. While this model correctly identified 1,404 enemy pairs, it indicated that an additional 71,285 pairs might be enemies, up from the 47,205 pairs of Model 3, but still a considerable improvement over the 289,249 pairs flagged by Model 2.

Table 21. Confusion Matrix, With JSD, After SME Feedback

		Actual: No	Actual: Yes	Total
Predicted: No	**Frequency**	25,879,551.00	170.00	**25,880,000.00**
	%	99.72	0.00	**99.72**
	Row %	100.00	0.00	**0.00**
	Column %	99.73	10.80	**0.00**
Predicted: Yes	**Frequency**	71,285.00	1,404.00	**72,689.00**
	%	0.27	0.01	**0.28**
	Row %	98.07	1.93	**0.00**
	Column %	0.27	89.20	**0.00**
Total	**Frequency**	**25,950,836.00**	**1,574.00**	**25,952,410.00**
	%	**99.99**	**0.01**	**100.00**

For the random forest classifier trained with the item pair JSD, and with SME input (Model 4), sensitivity improved considerably from Model 3's 0.78 to 0.89, or 78% correct classification of enemy pairs (Table 22). This degree of specificity is quite close to the best specificity found across all four models, 0.90 of Model 2. In addition, specificity and accuracy both maintain values 1.00 each, also found in Models 1 and 3.

Table 22. Classifier Performance, With JSD, With SME Feedback

Sensitivity	Specificity	Accuracy
0.89	1.00	1.00

Again, ROC curves may be used to provide a concise visual comparison of classifiers (Figure 11). Model 2 and Model 3 perform roughly equivalently, suggesting that SME input offers as much of an advantage as the inclusion of Jensen-Shannon distances. Model 4, which uses both the improved enemy information as well as the Jensen-Shannon distances, outperforms all other models.

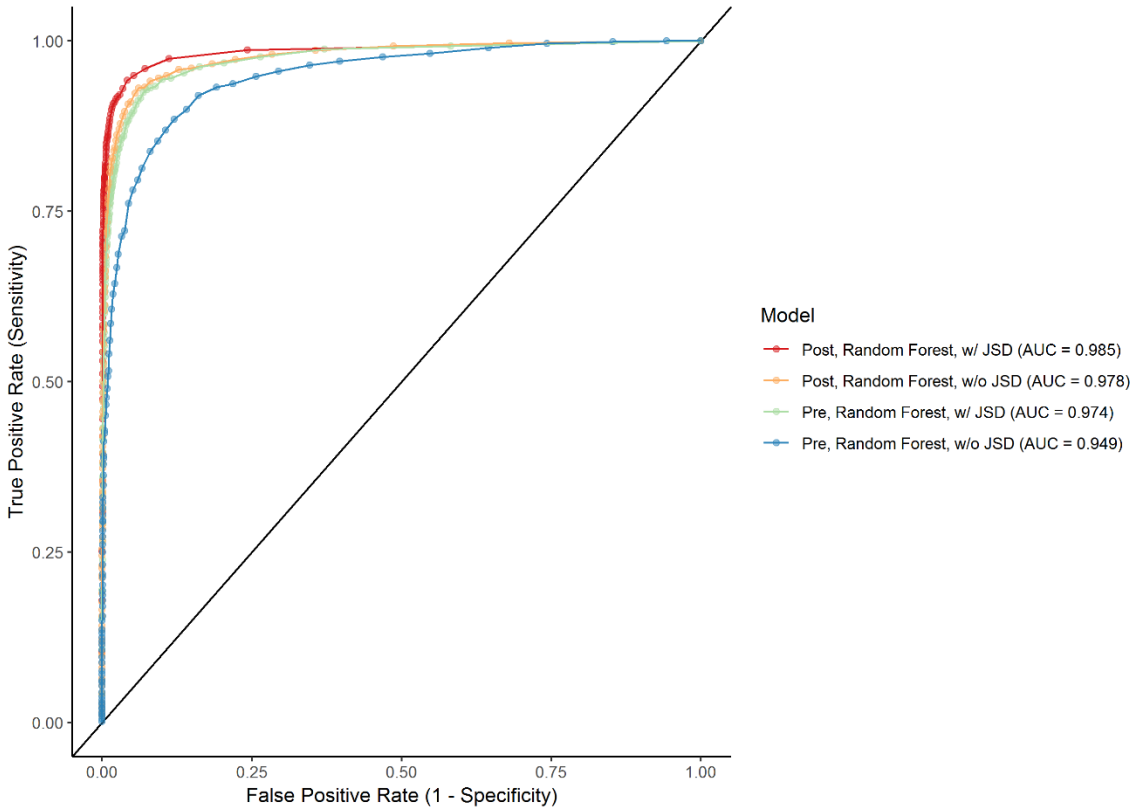


Figure 11. Empirical ROC Curves, Before and After SME Feedback

CHAPTER V

IMPLICATIONS, LIMITATIONS, AND FUTURE RESEARCH

In the following few pages, the implications of this study will be discussed, especially as they relate to the operationalization of the methods described in chapters three and four. In addition, the limitations of this study will be addressed, as well as directions future research might take.

Implications

The following section addresses the implications of this study, with attention to the operationalization of the methods described in this study. In particular, annual and semi-annual bank maintenance steps will be discussed, as well as the implications for automated test assembly procedures.

Annual Maintenance

As item writing committees submit new items, ostensibly on an annual schedule, those items may be scored by the latent Dirichlet allocation model. This will automatically generate *gamma* distributions for all new items, from which Jensen-Shannon distances may be calculated between these new items and all existing items in the bank. The benefit of this approach is that all new items can almost instantly be processed, and reasonable estimates of their enemies can be identified in the item bank.

In addition to processing new items on an annual basis, item pairs that have been identified by the random forest model, but that have not been assessed by subject matter experts (SMEs), can be served to SMEs via the app that was developed as part of this study.

In practical terms, this might look like the following: a committee of SMEs meets for annual form review. The committee is broken into groups that work on specific forms; as groups complete their work, instead of sitting by idly while others finish, they can simply navigate to the enemy-pair application and classify those pairs. Testing companies convene committees at great expense, and committee members, generally speaking, don't like sitting around with nothing to do. This is one way to capitalize on down time, to make committee members feel like they're contributing even if it is in ten-minute chunks.

As has been shown in this study, this gold-standard feedback from content experts can have a marked impact on the machine-learning models that try to predict enemy relationships. Providing feedback to the random forest model about pairs that it identified/flagged as enemies but in fact are not improves the model's ability to identify enemy item pairs. Remember, the model as it was trained at the outset of this study assumed that any item pairs that were not labeled as enemies *were not enemies*. However, because of the way enemy pairs are coded, this is not at all the case, and consequently the random forest model suffers. A program of continued, low intensity effort to assess item pairs about which the random forest model is highly confident is the surest and quickest way to train the model to more accurately identify enemy item pairs.

Having assessed those item pairs, the new enemy item classifications can be loaded into the item bank, and the random forest model can be retrained, revalidated, and the item bank can be rescored. The effect of this will be to have two enemy item fields in any given item bank: one might be a vector of comma-separated item IDs that are enemies of a given item, as identified by SMEs; the other field might be a vector of comma-separated item IDs that are *likely* enemies of a given item, as categorized by the random forest model. The first of these two fields is a gold-standard, expert-driven classification; these items must not appear on the same form. The second field is one that may be used when banks are healthy and can compensate for tighter constraints on test construction; excluding these items ought to reduce the burden on SMEs during form review meetings.

Semi-annual Maintenance

Annually, as new items are submitted to the item bank, they are scored by the LDA model and the random forest model as discussed above. On a semi-annual schedule, it is prudent to retrain the LDA model. As the bank grows, the corpus changes, and the topics themselves change. Over long stretches of time, topical changes are obvious: standards of care in the Middle Ages suggested that bloodletting via leeches might bring the body's four humours into balance and thereby restore health. A medical credentialing exam of the period might have been populated by topics related to bloodletting and the four humours; however, these topics are nowhere to be found in contemporary item banks. Similarly, in the 1980s, topics related to opioid addiction were virtually absent whereas today, given the current opioid crisis, it would not be surprising

to find items related to this topic on any given exam about the practice of general medicine.

Retraining the LDA topic model periodically accounts for shifts in standards of care and ensures that estimates of item-to-item proximity are based on a realistic snapshot of a fluid corpus.

Limitations of the Study

The limitations discussed in the following paragraphs are all related, in one way or another, to the data source. This study uses a secure, operational item bank, and it is, therefore, a study that is not reproducible by other researchers outside of the organization. Psychological research published in peer reviewed American Psychological Association (APA) periodicals generally are *not* reproducible because the researchers are unable or unwilling to share their data with other researchers (Wicherts, Borsboom, Kats, & Molenaar, 2006). This is an issue, because replication by independent researchers is a—if not *the*—gold standard by which the larger research community may judge the quality and validity of hypotheses and claims. This might be particularly true in an age of machine learning research, as many of these algorithms are "black box" in nature; the decision trees in a random forest model cannot be generalized into a simple, one-line formula into which other data may be plugged. Because of these under-the-hood complexities, independent researchers training their own models on common data is perhaps the best way to validate findings. It is unfortunate that this study is yet another that is not easily replicable because the researcher is unable to share the dataset at the core of the study. However, while full reproducibility is not possible here, as other

researchers continue to study the automated identification of enemy item pairs, the methods proposed herein, as well as the code and libraries used to conduct the study, might be shared as a middle-ground solution (Peng, 2011).

Other limitations of this study are rooted in the interaction between the bank and the models. For instance, because test items are such short documents, they can be challenging in a topic modeling context. In the case of this study, after stemming and removing stop words, there were fewer unique words than documents. In more traditional topic modeling contexts, where whole articles, book, or movies might be the objects of the model, the topics may be more stable. However, an eye should be kept on research in other contexts where topic modeling is being brought to bear, such as in studies of micro blogging, tweeting, and texting. Methods will undoubtedly evolve.

The nature of the data set used in this study is such that all documents focus on the same general context—that is, medical knowledge and one's ability to serve safely as a medical professional. This means that there is a relative lack of diversity in the terms and topics found in the corpus. In the *Time* Magazine Corpus (Davies, 2007), while there is a similar limitation imposed by a middle-class, educated audience, there is a vast array of terms and topics, ranging from the arts to politics to pop culture, especially by comparison with a medical licensing exam. This lack of diversity makes it more difficult to find the nuanced distinctions between discrete topics in an item bank.

Future Research

Beyond research into refinements of the methods described in this study, as well as the long-term analysis of how the models evolve over time and their contributions to

operational testing, this study has revealed other areas of research that might be pursued. The following section describes two areas of future research.

Automated Test Assembly and Random Forest Classifier Estimates

As described briefly above, an item bank under this proposed paradigm will have two enemy item fields: the first will be a vector of comma-separated item IDs that are enemies of a given item, as identified by SMEs; these are expert-driven classifications indicating items must not appear on the same form.

A second field will be a vector populated by comma-separated item IDs that are *likely* enemies of a given item as categorized by the random forest model. Remember that the random forest model estimates a probability that any two items are enemies; all item pairs were assigned a dichotomous classification of Enemies or Not Enemies. For the assessment of the models (confusion matrices and ROC curves), a probability greater than 0.5 resulted in a classification of Enemies. However, recall that items presented to SMEs were selected only if they had a probability of 0.9 or higher. That is, the threshold for classification was considerably higher. This second field might be populated in the same manner, with a higher threshold.

An automated test assembly (ATA) algorithm, then, might select items based on the SME-generated enemy classifications, as well as the estimated enemy classifications. A healthier bank might allow for a higher bar in terms of an enemy classification threshold, while a more anemic bank might only use the enemy item classifications generated by SMEs. The goal of this approach—indeed, the goal of this entire study—is to weed out as many enemy pairs from forms *before* forms review, thereby reducing the

burden on SMEs. A simple study of the number of item replacements on forms as a result of enemy item interactions before and after the implementation of such a system would be a terrific initial step toward evaluating the impact of automated enemy detection.

Partner Item Identification

Consider a longitudinal exam program that seeks to assess knowledge and to encourage growth among the examinees. In this scenario, examinees are alerted to incorrect answers and given time to study material before resuming the exam. At some point in the future, examinees can expect to see items related to some of those they previously answered incorrectly—the items will be selected for the examinee specifically to reassess these knowledge deficits.

The items chosen for reassessment are known as partner items. Partner items are useful enemies, as they target the same material as another item without being a clone of that item. Identifying likely partner items in a test bank with methods like those outlined in this study would save considerable time, effort, and money.

REFERENCES

Ackerman, T. A., & Spray, J. A. (1986). A General Model for Item Dependency. In *67th Annual Meeting of the American Educational Research Association*. San Francisco. Retrieved from https://eric.ed.gov/?id=ED272579

American Educational Research Association, American Psychological Association, & National Council on Measurement in Education. (2014). *Standards for educational and psychological testing*. Washington, DC: AERA.

Arun, R., Suresh, V., Veni Madhavan, C. E., & Narasimha Murthy, M. N. (2010). On finding the Natural Number of Topics with Latent Dirichlet Allocation: Some observations. In *Advances in Knowledge Discovery and Data Mining* (pp. 391–402). Springer, Berlin, Heidelberg. Retrieved from http://link.springer.com/10.1007/978-3-642-13657-3_43

Becker, K. A., & Kao, S.-C. (2009). Finding stolen items and improving item banks. In *American Educational Research Association Annual Meeting*. San Diego, CA.

Blei, D. M. (2012). Probabilistic topic models. *Communications of the ACM*, *55*(4), 77–84. https://doi.org/10.1145/2133806.2133826

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, *3*, 933–1022.

Bouchet-Valet, M. (2014). SnowballC. Retrieved from https://r-forge.r-project.org/projects/r-temis/

Breiman, L. (1996). Bagging predictors. *Machine Learning*, *24*(2), 123–140. https://doi.org/10.1007/BF00058655

Breiman, L. (2001a). Random forests. *Machine Learning*, *45*(1), 5–32.

Breiman, L. (2001b). Random Forests. *Machine Learning*, *45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

Brown, P. F., Della Pietra, S. A., Della Pietra, V. J., Lai, J. C., & Mercer, R. L. (1992). An estimate of an upper bound for the entropy of English. *Computational Linguistics*, *18*(1), 31–40.

Buckley, C. (1985). *Implementation of the SMART information retrieval system*. Ithaca, New York: Cornell University.

Cao, J., Xia, T., Li, J., Zhang, Y., & Tang, S. (2009). A density-based method for adaptive LDA model selection. *Neurocomputing*, *72*(7–9), 1775–1781. https://doi.org/10.1016/J.NEUCOM.2008.06.011

Case, S. M., & Swanson, D. B. (1998). *Constructing written test questions for the basic and clinical sciences* (3rd ed.). Philadelphia, PA: National Board of Medical Examiners.

Chowdhury, G. G. (2003). Natural language processing. *Annual Review of Information Science and Technology*, *37*(1), 51–89. https://doi.org/10.1002/aris.1440370103

Cook, T. D., & Campbell, D. T. (1979). *Quasi-experimentation: Design and analysis issues for field settings* (Vol. 3). Chicago: Rand McNally.

Davies, M. (2007). Time Magazine Corpus: 100 million words, 1920s - 2000s. Brigham Young University. Retrieved from https://corpus.byu.edu/time/

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science.*, *41*(6).

Downing, S. M., & Haladyna, T. M. (2006). *Handbook of test development*. L. Erlbaum.

Downing, S. M., & Haladyna, T. M. (Eds.). (2013). *Handbook of Test Development*. Routledge. https://doi.org/10.4324/9780203874776

Drasgow, F., Luecht, R. M., & Bennett, R. (2006). Technology and testing. In R. L. Brennan (Ed.), *Educational Measurememement* (4th ed., pp. 471–516). Westport, CT.

Embretson, S. E., & Reise, S. P. (2000). *Item response theory for psychologists*. L. Erlbaum Associates.

Gierl, M. J., & Haladyna, T. M. (2013). *Automatic item generation: theory and practice*. New York NY: Routledge. Retrieved from http://uncg.worldcat.org.libproxy.uncg.edu/title/automatic-item-generation-theory-and-practice/oclc/754731188&referer=brief_results

Goodman, J. (2008). *An examination of the residual covariance structures of complex performance exercises under various scaling and scoring methods*. University of North Carolina at Greensboro.

Griffiths, T. L., & Steyvers, M. (2004). Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America*, *101 Suppl 1*(suppl 1), 5228–5235. https://doi.org/10.1073/pnas.0307752101

Haladyna, T. M., & Downing, S. M. (2004). Construct-irrelevant variance in high-stakes testing. *Educational Measurement: Issues and Practice*, *23*(1), 17–27.

Haladyna, T. M., & Shindoll, R. R. (1989). Item shells: A method for writing effective multiple-choice test iterms. *Evaluation & the Health Professions*, *12*(1), 97–106. Retrieved from http://journals.sagepub.com/doi/10.1177/016327878901200106

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning*. New York, NY: Springer New York. https://doi.org/10.1007/978-0-387-84858-7

Huitzing, H., Veldkamp, B. P., & Verschoor, A. J. (2005). Infeasibility in automated test assembly models: A comparison study of different methods. *Journal of Educational Measurement*, *42*(3).

Jain, A. K. (2010). Data clustering: 50 years beyond K-means. *Pattern Recognition Letters*, *31*(8), 651–666. Retrieved from http://linkinghub.elsevier.com/retrieve/pii/S0167865509002323

Kane, M. (1992). An argument-based approach to validity. *Psychological Bulletin*, *112*(3).

Kane, M. (2004). Certification testing as an illustration of argument-based validation. *Measurement*. Retrieved from http://www.tandfonline.com/doi/abs/10.1207/s15366359mea0203_1

Kane, M. (2006). Validation. In R. L. Brennan (Ed.), *Educational measurement* (4th ed., pp. 17–64). Westport, CT: Praeger Publishers.

Kane, M. (2013). Validity and fairness in the testing of individuals. In M. Chatterji (Ed.), *Validity and test use: An international dialogue on educational assessment, accountability and equity* (1st ed.). Bingley, UK: Emerald.

Lai, H., & Becker, K. A. (2010). Detecting enemy item pairs using Artificial Neural Networks. In *National Council on Measurement in Education Annual Meeting*. Denver, CO.

Lord, F. M., & Novick, M. R. (1968). *Statistical theories of mental test scores*. Reading, MA: Addison-Wesley.

Luecht, R. M., & Sireci, S. G. (2011). *A Review of Models for Computer-Based Testing*. New York.

Mellenbergh, G. J. (1994). Generalized linear item response theory. *Psychological Bulletin*, *115*(2), 300–307. https://doi.org/10.1037/0033-2909.115.2.300

Messick, S. (1989). Validity. In R. L. Linn (Ed.), *Eduational measurement.* (3rd ed., pp. 13–103). New York: Macmillan.

Onix text retrieval toolkit: Stopword list 1. (n.d.). Retrieved February 28, 2019, from https://www.lextek.com/manuals/onix/stopwords1.html

Papadimitriou, C. H., Raghavan, P., Tamaki, H., & Vempala, S. (2000). Latent Semantic Indexing: A Probabilistic Analysis. *Journal of Computer and System Sciences*, *61*(2), 217–235. https://doi.org/10.1006/jcss.2000.1711

Peng, R. D. (2011). Reproducible research in computational science. *Science*, *334*(6060). Retrieved from http://science.sciencemag.org/content/334/6060/1226.short

Polikar, R. (2012). Ensemble learning. In C. Zhang & Y. Ma (Eds.), *Ensemble machine learning: Methods and applications* (pp. 1–34). New York: Springer.

Pommerich, M., & Segall, D. O. (2008). Local dependence in an operational CAT: Diagnosis and implications. *Journal of Educational Measurement*, *45*(3), 201–223.

Porter, M. F. (n.d.). Snowball: A language for stemming algorithms. Retrieved February 28, 2019, from http://snowballstem.org/texts/introduction.html

Porter, M. F. (1980). An algorithm for suffix stripping. *Program*, *40*(3), 211–218. https://doi.org/10.1108/00330330610681286

Rajaraman, A., & Ullman, J. D. (2011). *Mining of Massive Datasets*. Cambridge: Cambridge University Press. https://doi.org/10.1017/CBO9781139058452

Silge, J., & Robinson, D. (2017). *Text Mining with R*. Sebastapol, CA: O'Reilly Media.

Tepsumethanon, V., Wilde, H., & Meslin, F. X. (2005). Six criteria for rabies diagnosis in living dogs. *Journal of the Medical Association of Thailand*, *88*(3), 419–422.

van der Linden, W. J. (2005). *Linear models for optimal test design*. New York NY: Springer.

Veldkamp, B. P. (2013). Application of robust optimization to automated test assembly. *Annals of Operations Research*, *206*(1).

Wicherts, J. M., Borsboom, D., Kats, J., & Molenaar, D. (2006). The poor availability of psychological research data for reanalysis. *American Psychologist*, *61*(7). Retrieved from https://psycnet.apa.org/doiLanding?doi=10.1037%2F0003-066X.61.7.726

Wilbur, W. J., & Sirotkin, K. (1992). The automatic identification of stop words. *Journal of Information Science*, *18*(1), 45–55. https://doi.org/10.1177/016555159201800106

Woo, A., & Gorham, J. (2010). Understanding the impact of enemy items on test validity and measurement precision. *CLEAR Exam Review*, *21*(1), 15–17.

Yen, W. M. (1984). Effects of local item dependence on the fit and equating performance of the three-parameter logistic model. *Applied Psychological Measurement*, *8*(2), 125–145. https://doi.org/10.1177/014662168400800201

Yen, W. M. (1993). Scaling performance assessments: Strategies for managing local item dependence. *Journal of Educational Measurement*, *30*(3), 187–213.

Zhang, C., & Ma, Y. (Eds.). (2012). *Ensemble machine learning: Methods and applications*. New York: Springer.

Zhou, Z.-H. (2012). *Ensemble methods: Foundations and algorithms*. Cambridge: CRC Press. Retrieved from http://uncg.worldcat.org/title/ensemble-methods-foundations-and-algorithms/oclc/796675544&referer=brief_results

APPENDIX A

BETA MATRIX SAMPLE

	Topic									
Term	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
abat	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
abdomen	0.000	0.000	0.000	0.001	0.000	0.000	0.000	0.000	0.000	0.001
abdomin	0.000	0.000	0.000	0.002	0.000	0.000	0.000	0.000	0.000	0.001
abduct	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.002
abductor	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
abil	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
abnorm	0.022	0.004	0.010	0.014	0.000	0.007	0.000	0.000	0.001	0.014
abo	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
abort	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
aborta	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
abras	0.000	0.000	0.000	0.000	0.001	0.000	0.000	0.000	0.000	0.000
abroad	0.000	0.000	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000
abrupt	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
abruptio	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
abruptli	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
abscess	0.000	0.000	0.000	0.000	0.009	0.000	0.000	0.000	0.000	0.000
absenc	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
absent	0.000	0.004	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
absolut	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001
absorb	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
absorpt	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
abstract	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
abus	0.000	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.003	0.000
academ	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.000
acalcul	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
acanthosi	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.000	0.000	0.000
acarbos	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.000	0.000
accentu	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
access	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
accessori	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
accid	0.000	0.000	0.000	0.000	0.005	0.000	0.000	0.000	0.000	0.000
accident	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
…						…				
zoster	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

	Topic									
Term	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
abat	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
abdomen	0.000	0.000	0.000	0.001	0.000	0.000	0.000	0.000	0.040	0.000
abdomin	0.000	0.000	0.000	0.000	0.001	0.002	0.000	0.000	0.017	0.000
abduct	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
abductor	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
abil	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.000	0.000	0.000
abnorm	0.007	0.006	0.000	0.027	0.008	0.002	0.002	0.003	0.005	0.000
abo	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
abort	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
aborta	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
abras	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
abroad	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
abrupt	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
abruptio	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
abruptli	0.000	0.000	0.000	0.000	0.000	0.001	0.000	0.000	0.000	0.000
abscess	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.001	0.000
absenc	0.000	0.002	0.000	0.000	0.000	0.000	0.000	0.002	0.000	0.000
absent	0.000	0.000	0.000	0.000	0.001	0.000	0.000	0.000	0.001	0.000
absolut	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
absorb	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
absorpt	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001
abstract	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
abus	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.003	0.000
academ	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
acalcul	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
acanthosi	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
acarbos	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
accentu	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
access	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
accessori	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
accid	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
accident	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
…						…				
zoster	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

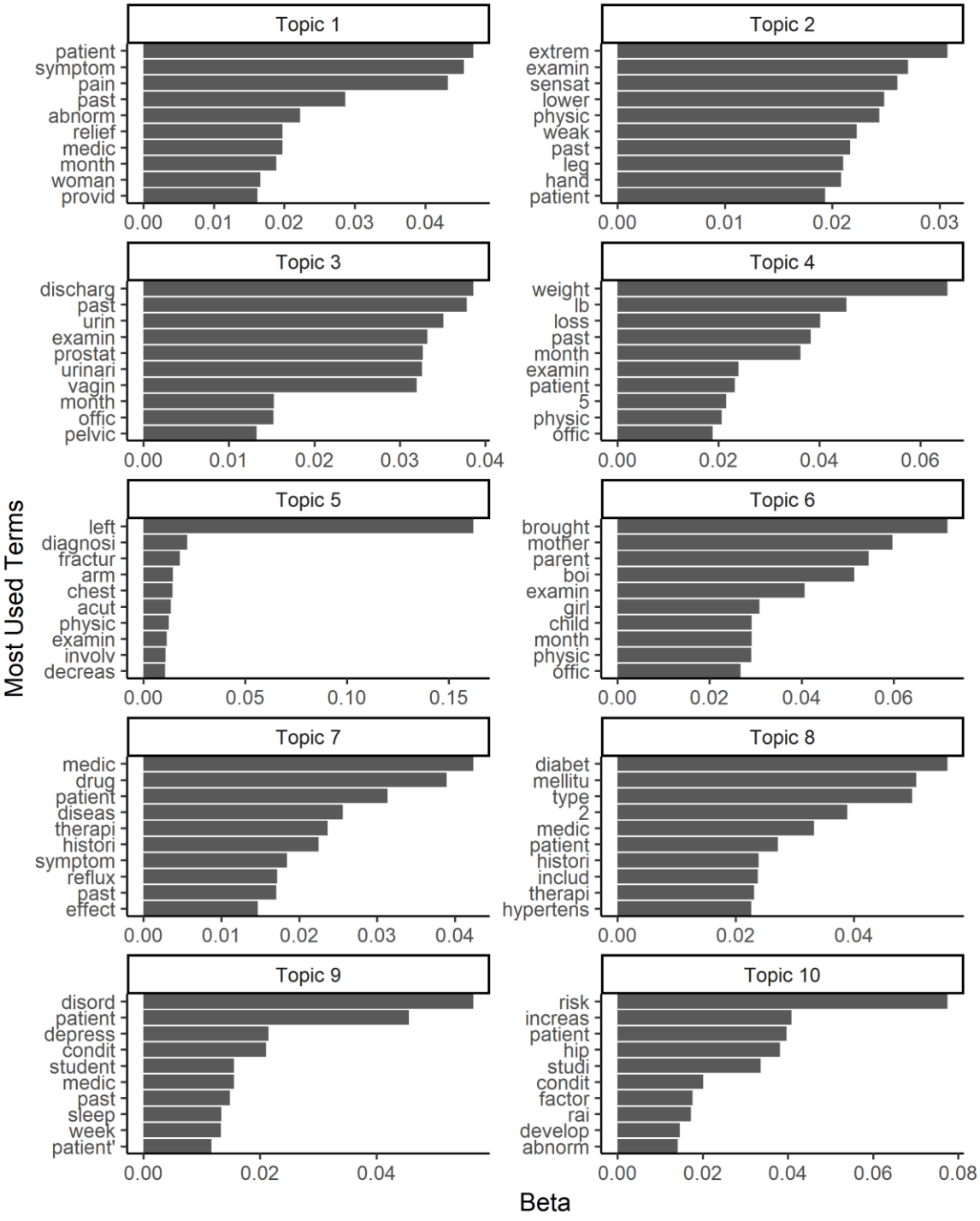
	Topics									
Term	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**
abat	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
abdomen	0.000	0.000	0.000	0.008	0.001	0.000	0.000	0.000	0.005	0.001
abdomin	0.000	0.000	0.000	0.002	0.000	0.000	0.000	0.000	0.000	0.000
abduct	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
abductor	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
abil	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.000	0.000	0.001
abnorm	0.001	0.001	0.006	0.008	0.003	0.004	0.018	0.000	0.000	0.000
abo	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
abort	0.000	0.000	0.002	0.000	0.000	0.000	0.000	0.000	0.000	0.000
aborta	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
abras	0.002	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
abroad	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
abrupt	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
abruptio	0.000	0.000	0.000	0.001	0.000	0.000	0.000	0.000	0.000	0.000
abruptli	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
abscess	0.000	0.000	0.000	0.000	0.000	0.001	0.000	0.000	0.000	0.000
absenc	0.000	0.000	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.001
absent	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
absolut	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.003
absorb	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
absorpt	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
abstract	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
abus	0.002	0.003	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
academ	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
acalcul	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
acanthosi	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
acarbos	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
accentu	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
access	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
accessori	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
accid	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
accident	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
…					…					
zoster	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

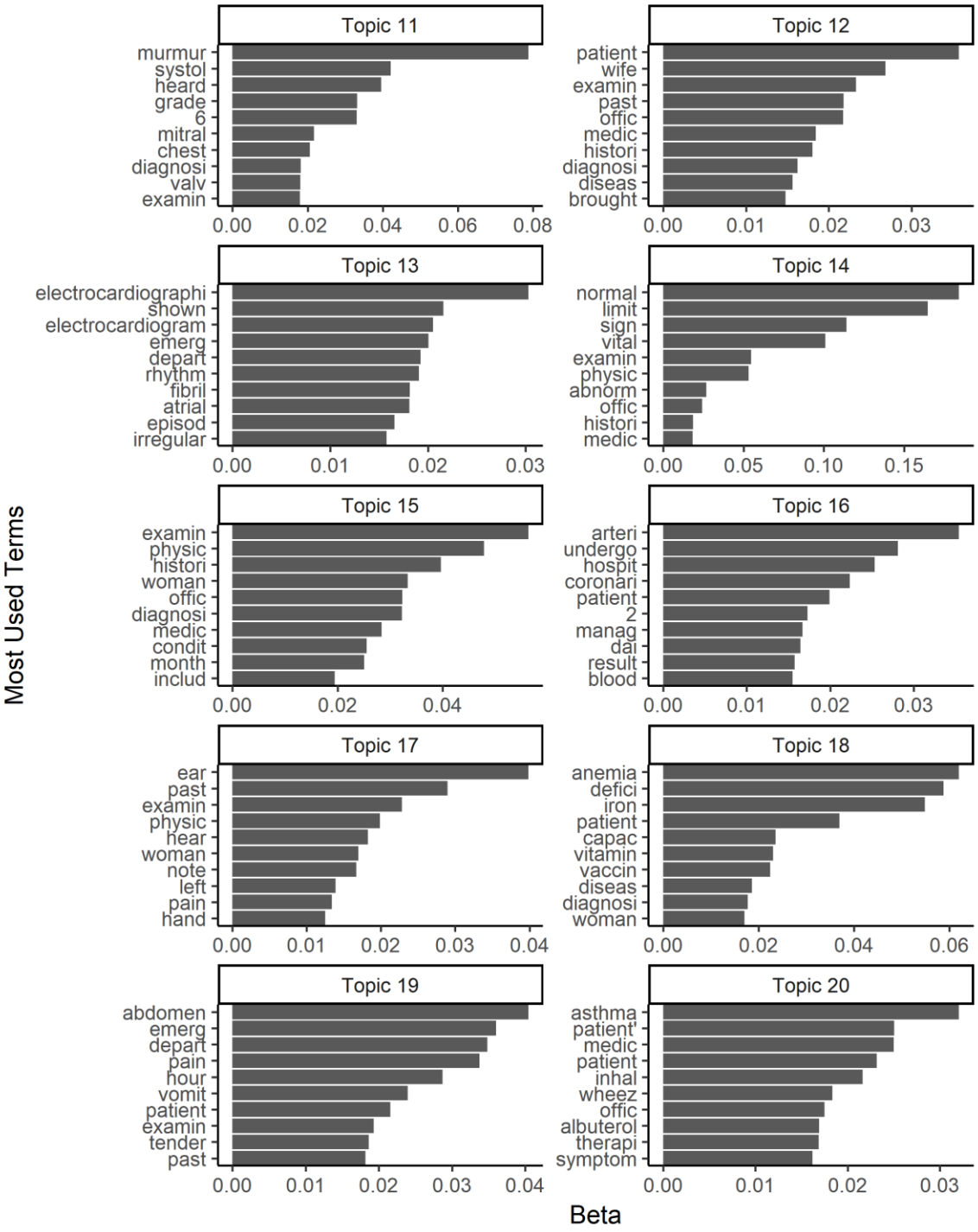
	Topic									
Term	**31**	**32**	**33**	**34**	**35**	**36**	**37**	**38**	**39**	**40**
abat	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
abdomen	0.000	0.000	0.000	0.007	0.000	0.000	0.000	0.001	0.001	0.046
abdomin	0.001	0.000	0.000	0.001	0.000	0.000	0.000	0.000	0.000	0.028
abduct	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
abductor	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
abil	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
abnorm	0.001	0.002	0.003	0.001	0.005	0.000	0.001	0.008	0.007	0.003
abo	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
abort	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
aborta	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
abras	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
abroad	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
abrupt	0.000	0.000	0.000	0.000	0.001	0.000	0.000	0.000	0.000	0.000
abruptio	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
abruptli	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
abscess	0.000	0.000	0.000	0.001	0.000	0.000	0.000	0.000	0.000	0.000
absenc	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001
absent	0.000	0.000	0.000	0.000	0.003	0.000	0.000	0.000	0.000	0.000
absolut	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
absorb	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
absorpt	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
abstract	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
abus	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
academ	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
acalcul	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.000	0.000
acanthosi	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
acarbos	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
accentu	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
access	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
accessori	0.000	0.000	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000
accid	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
accident	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
…						…				
zoster	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.253	0.000	0.000

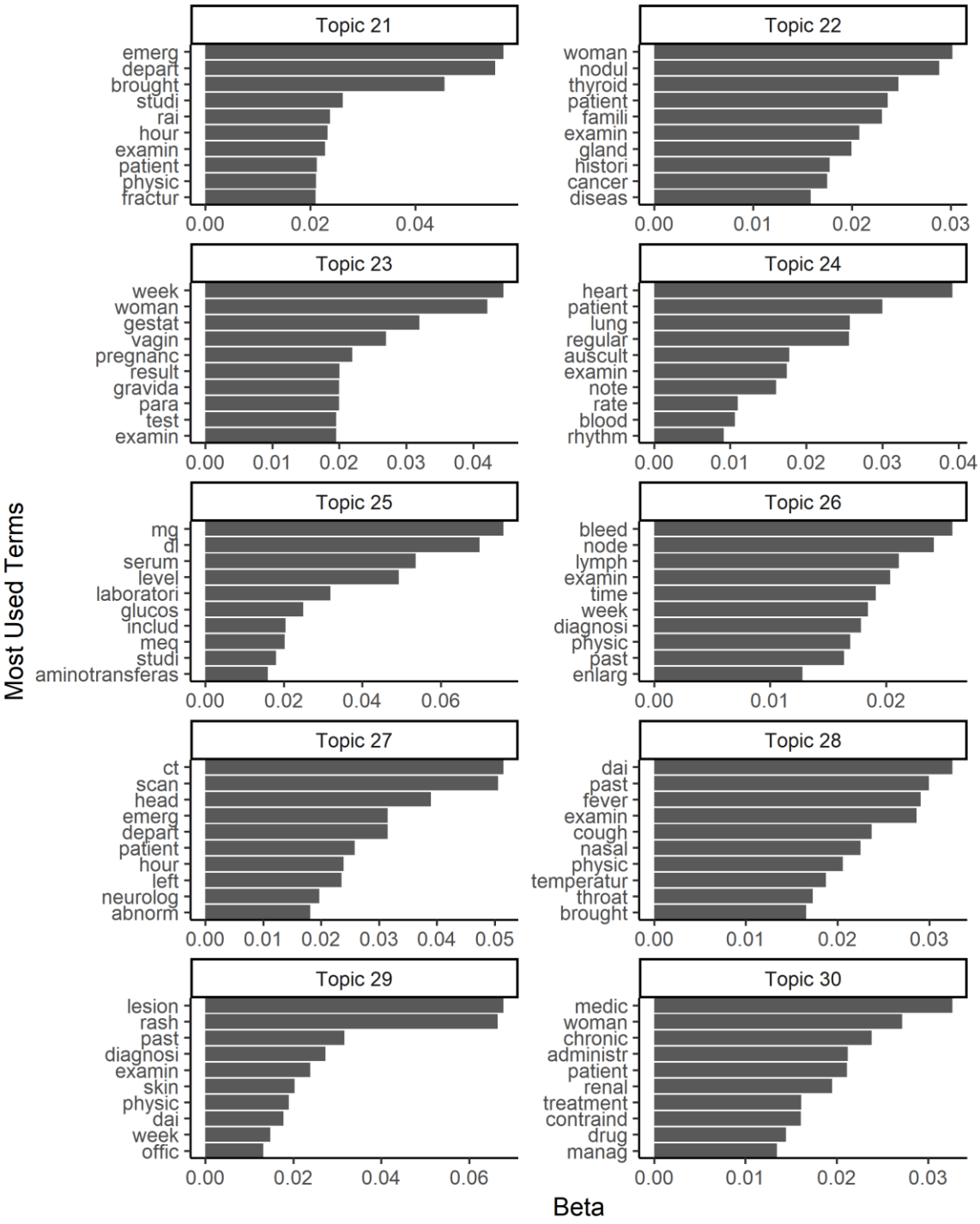
	Topic									
Term	**41**	**42**	**43**	**44**	**45**	**46**	**47**	**48**	**49**	**50**
abat	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
abdomen	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.011	0.000	0.000
abdomin	0.000	0.002	0.000	0.000	0.000	0.000	0.002	0.006	0.000	0.000
abduct	0.000	0.000	0.000	0.000	0.000	0.005	0.000	0.000	0.000	0.000
abductor	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
abil	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
abnorm	0.000	0.005	0.002	0.008	0.000	0.002	0.000	0.000	0.003	0.002
abo	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
abort	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
aborta	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
abras	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.000	0.001	0.000
abroad	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
abrupt	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
abruptio	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
abruptli	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
abscess	0.000	0.000	0.000	0.000	0.000	0.000	0.002	0.000	0.000	0.000
absenc	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.002	0.000
absent	0.000	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.002	0.000
absolut	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
absorb	0.000	0.000	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000
absorpt	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
abstract	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
abus	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
academ	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
acalcul	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
acanthosi	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
acarbos	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
accentu	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
access	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
accessori	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
accid	0.000	0.000	0.000	0.000	0.004	0.000	0.000	0.000	0.000	0.000
accident	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.000	0.000	0.000
…					…					
zoster	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

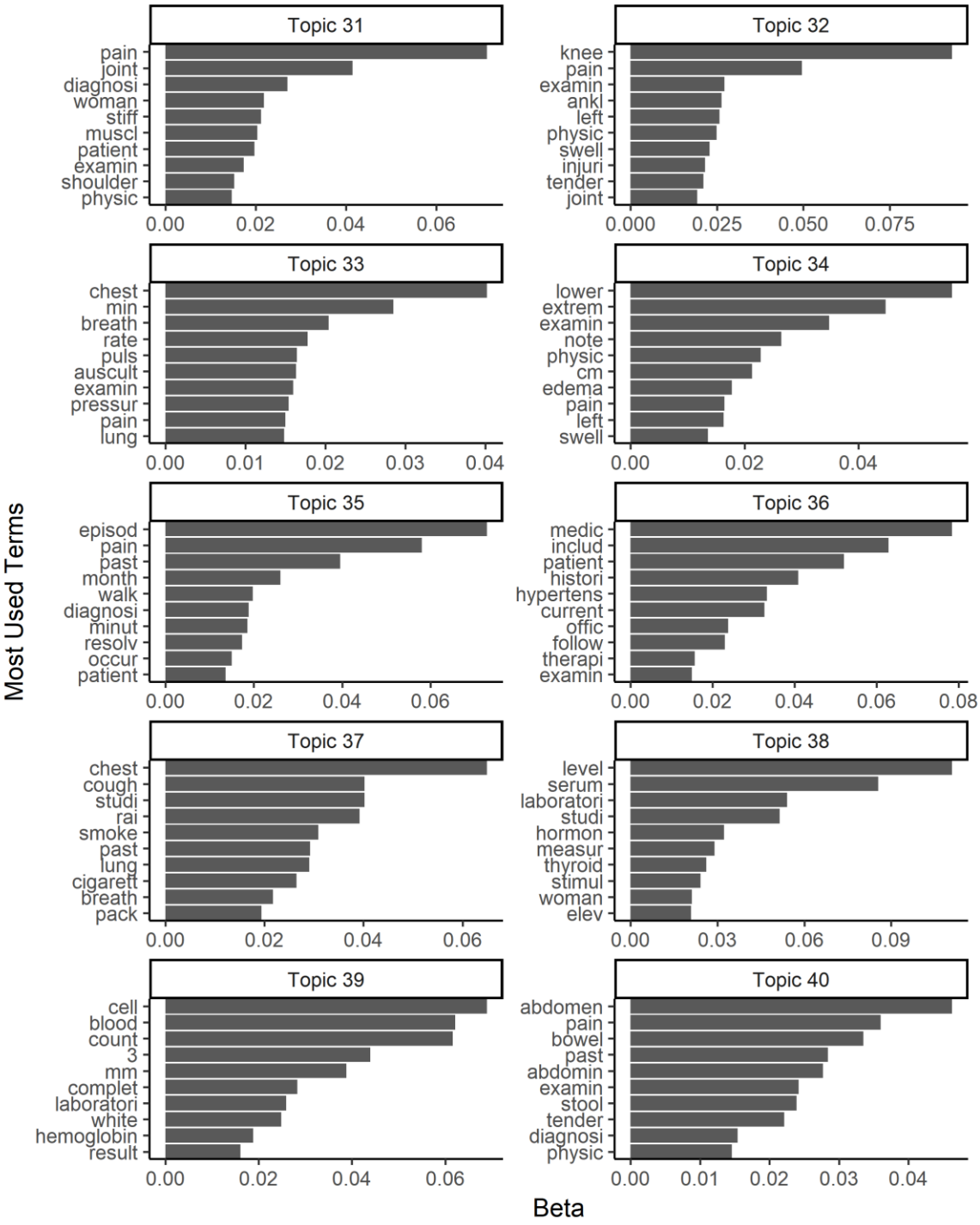
APPENDIX B

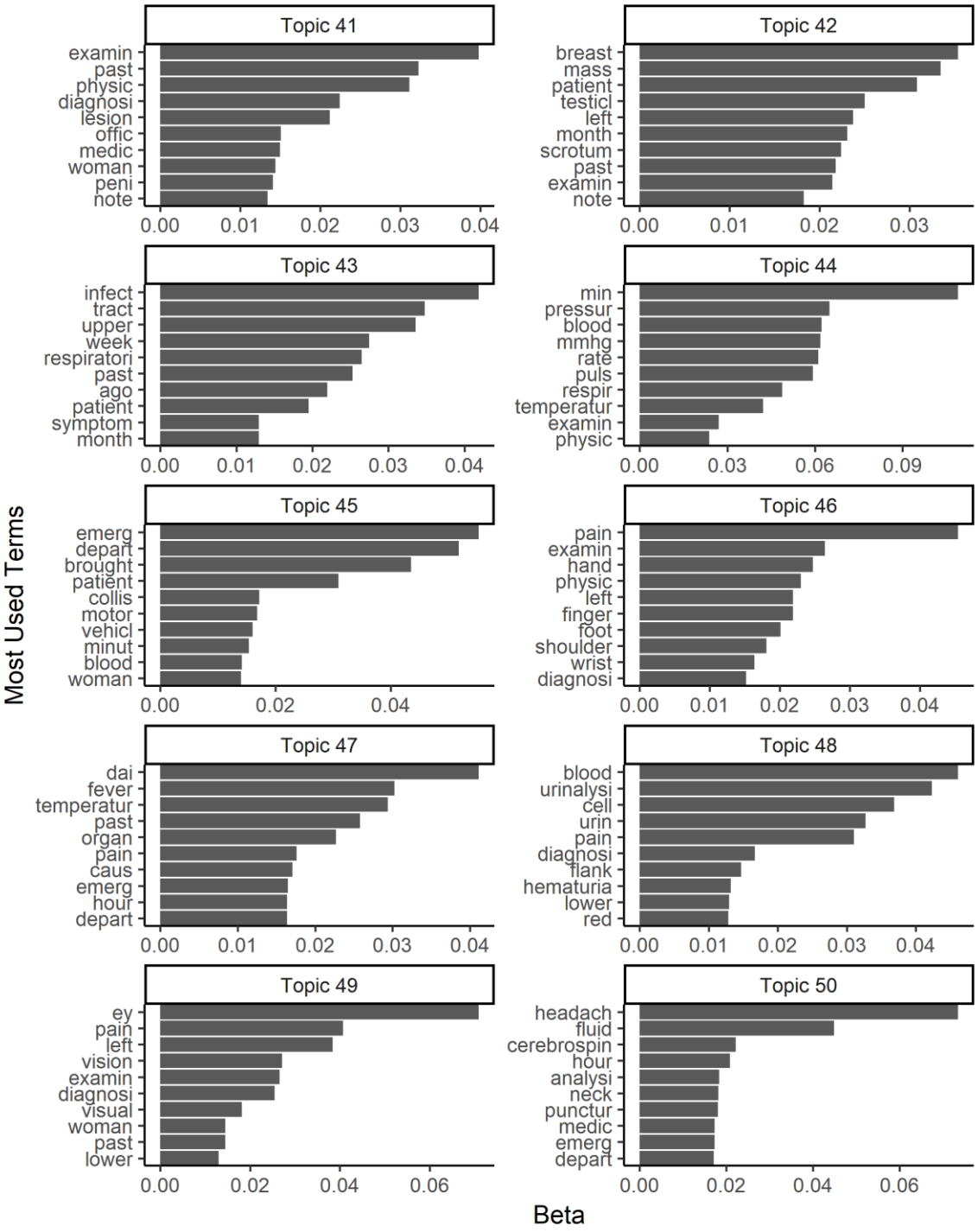
MOST COMMON WORDS PER TOPIC




Topic 11
murmur
systol
heard
grade
6
mitral
chest
diagnosi
valv
examin
0.00
0.02
0.04
0.06
0.08
Topic 12
patient
wife
examin
past
offic
medic
histori
diagnosi
diseas
brought
0.00
0.01
0.02
0.03
Topic 13
electrocardiographi
shown
electrocardiogram
emerg
depart
rhythm
fibril
atrial
episod
irregular
0.00
0.01
0.02
0.03
Topic 14
normal
limit
sign
vital
examin
physic
abnorm
offic
histori
medic
0.00
0.05
0.10
0.15
Topic 15
examin
physic
histori
woman
offic
diagnosi
medic
condit
month
includ
0.00
0.02
0.04
Topic 16
arteri
undergo
hospit
coronari
patient
2
manag
dai
result
blood
0.00
0.01
0.02
0.03
Topic 17
ear
past
examin
physic
hear
woman
note
left
pain
hand
0.00
0.01
0.02
0.03
0.04
Topic 18
anemia
defici
iron
patient
capac
vitamin
vaccin
diseas
diagnosi
woman
0.00
0.02
0.04
0.06
Topic 19
abdomen
emerg
depart
pain
hour
vomit
patient
examin
tender
past
0.00
0.01
0.02
0.03
0.04
Topic 20
asthma
patient'
medic
patient
inhal
wheez
offic
albuterol
therapi
symptom
0.00
0.01
0.02
0.03
Most Used Terms
Beta


Topic 21
emerg
depart
brought
studi
rai
hour
examin
patient
physic
fractur
0.00
0.02
0.04
Topic 22
woman
nodul
thyroid
patient
famili
examin
gland
histori
cancer
diseas
0.00
0.01
0.02
0.03
Topic 23
week
woman
gestat
vagin
pregnanc
result
gravida
para
test
examin
0.00
0.01
0.02
0.03
0.04
Topic 24
heart
patient
lung
regular
auscult
examin
note
rate
blood
rhythm
0.00
0.01
0.02
0.03
0.04
Topic 25
mg
dl
serum
level
laboratori
glucos
includ
meq
studi
aminotransferas
0.00
0.02
0.04
0.06
Topic 26
bleed
node
lymph
examin
time
week
diagnosi
physic
past
enlarg
0.00
0.01
0.02
Topic 27
ct
scan
head
emerg
depart
patient
hour
left
neurolog
abnorm
0.00
0.01
0.02
0.03
0.04
0.05
Topic 28
dai
past
fever
examin
cough
nasal
physic
temperatur
throat
brought
0.00
0.01
0.02
0.03
Topic 29
lesion
rash
past
diagnosi
examin
skin
physic
dai
week
offic
0.00
0.02
0.04
0.06
Topic 30
medic
woman
chronic
administr
patient
renal
treatment
contraind
drug
manag
0.00
0.01
0.02
0.03
Most Used Terms
Beta


Topic 31
pain
joint
diagnosi
woman
stiff
muscl
patient
examin
shoulder
physic
0.00
0.02
0.04
0.06
Topic 32
knee
pain
examin
ankl
left
physic
swell
injuri
tender
joint
0.000
0.025
0.050
0.075
Topic 33
chest
min
breath
rate
puls
auscult
examin
pressur
pain
lung
0.00
0.01
0.02
0.03
0.04
Topic 34
lower
extrem
examin
note
physic
cm
edema
pain
left
swell
0.00
0.02
0.04
Topic 35
episod
pain
past
month
walk
diagnosi
minut
resolv
occur
patient
0.00
0.02
0.04
0.06
Topic 36
medic
includ
patient
histori
hypertens
current
offic
follow
therapi
examin
0.00
0.02
0.04
0.06
0.08
Topic 37
chest
cough
studi
rai
smoke
past
lung
cigarett
breath
pack
0.00
0.02
0.04
0.06
Topic 38
level
serum
laboratori
studi
hormon
measur
thyroid
stimul
woman
elev
0.00
0.03
0.06
0.09
Topic 39
cell
blood
count
3
mm
complet
laboratori
white
hemoglobin
result
0.00
0.02
0.04
0.06
Topic 40
abdomen
pain
bowel
past
abdomin
examin
stool
tender
diagnosi
physic
0.00
0.01
0.02
0.03
0.04
Most Used Terms
Beta


Topic 41
examin
past
physic
diagnosi
lesion
offic
medic
woman
peni
note
0.00
0.01
0.02
0.03
0.04
Topic 42
breast
mass
patient
testicl
left
month
scrotum
past
examin
note
0.00
0.01
0.02
0.03
Topic 43
infect
tract
upper
week
respiratori
past
ago
patient
symptom
month
0.00
0.01
0.02
0.03
0.04
Topic 44
min
pressur
blood
mmhg
rate
puls
respir
temperatur
examin
physic
0.00
0.03
0.06
0.09
Topic 45
emerg
depart
brought
patient
collis
motor
vehicl
minut
blood
woman
0.00
0.02
0.04
Topic 46
pain
examin
hand
physic
left
finger
foot
shoulder
wrist
diagnosi
0.00
0.01
0.02
0.03
0.04
Topic 47
dai
fever
temperatur
past
organ
pain
caus
emerg
hour
depart
0.00
0.01
0.02
0.03
0.04
Topic 48
blood
urinalysi
cell
urin
pain
diagnosi
flank
hematuria
lower
red
0.00
0.01
0.02
0.03
0.04
Topic 49
ey
pain
left
vision
examin
diagnosi
visual
woman
past
lower
0.00
0.02
0.04
0.06
Topic 50
headach
fluid
cerebrospin
hour
analysi
neck
punctur
medic
emerg
depart
0.00
0.02
0.04
0.06
Most Used Terms
Beta

APPENDIX C

GAMMA MATRIX SAMPLE

	Topic									
Document	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001
2	0.000	0.000	0.000	0.267	0.000	0.000	0.000	0.000	0.320	0.000
3	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
4	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
5	0.356	0.000	0.000	0.000	0.000	0.074	0.000	0.000	0.000	0.000
6	0.329	0.000	0.000	0.000	0.000	0.071	0.000	0.000	0.000	0.000
7	0.000	0.000	0.000	0.000	0.000	0.065	0.000	0.000	0.000	0.000
8	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001
9	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001
10	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
11	0.000	0.000	0.000	0.000	0.000	0.056	0.000	0.000	0.000	0.000
12	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
13	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.056
14	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.078	0.000	0.000
15	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.220
16	0.130	0.059	0.000	0.000	0.000	0.000	0.000	0.000	0.164	0.000
17	0.000	0.000	0.000	0.234	0.000	0.202	0.000	0.000	0.000	0.000
18	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
19	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.558	0.000	0.000
20	0.139	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
21	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
22	0.203	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
23	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
24	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
25	0.000	0.000	0.000	0.000	0.000	0.028	0.000	0.198	0.000	0.000
26	0.000	0.000	0.000	0.000	0.000	0.000	0.334	0.000	0.113	0.000
27	0.000	0.000	0.000	0.171	0.000	0.000	0.000	0.251	0.000	0.000
28	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
29	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
30	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.274	0.000
31	0.000	0.000	0.000	0.000	0.000	0.071	0.000	0.000	0.022	0.000
32	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.318	0.000	0.000
…						…				
7205	0.000	0.000	0.379	0.000	0.000	0.000	0.000	0.000	0.000	0.000

	Topic									
Document	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
1	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001
2	0.000	0.000	0.000	0.104	0.000	0.000	0.000	0.000	0.000	0.000
3	0.000	0.000	0.000	0.207	0.000	0.000	0.000	0.000	0.000	0.335
4	0.000	0.000	0.000	0.158	0.000	0.000	0.000	0.000	0.000	0.539
5	0.000	0.000	0.000	0.025	0.000	0.000	0.000	0.000	0.000	0.147
6	0.000	0.000	0.000	0.026	0.000	0.000	0.000	0.000	0.000	0.116
7	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.248
8	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001
9	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001
10	0.000	0.000	0.000	0.032	0.000	0.254	0.000	0.000	0.000	0.000
11	0.239	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
12	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
13	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001
14	0.000	0.020	0.000	0.000	0.000	0.000	0.285	0.000	0.000	0.000
15	0.000	0.000	0.000	0.319	0.000	0.000	0.000	0.000	0.000	0.000
16	0.247	0.000	0.181	0.000	0.000	0.000	0.000	0.000	0.000	0.000
17	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
18	0.070	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
19	0.000	0.000	0.000	0.000	0.000	0.421	0.000	0.000	0.000	0.000
20	0.000	0.000	0.000	0.023	0.000	0.000	0.000	0.000	0.000	0.000
21	0.000	0.000	0.000	0.078	0.000	0.144	0.000	0.000	0.000	0.229
22	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.470
23	0.000	0.000	0.000	0.259	0.000	0.000	0.000	0.000	0.000	0.000
24	0.000	0.000	0.000	0.200	0.000	0.000	0.000	0.000	0.000	0.000
25	0.000	0.000	0.000	0.173	0.000	0.000	0.000	0.000	0.000	0.000
26	0.000	0.000	0.360	0.000	0.000	0.000	0.000	0.000	0.000	0.000
27	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
28	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.082	0.000
29	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
30	0.000	0.167	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
31	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
32	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
…						…				
7205	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

	Topic									
Document	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**
1	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001
2	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
3	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
4	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
5	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
6	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.162	0.000
7	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.216	0.000
8	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001
9	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001
10	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.099
11	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
12	0.000	0.049	0.000	0.000	0.000	0.000	0.066	0.000	0.000	0.322
13	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001
14	0.000	0.000	0.000	0.000	0.043	0.000	0.069	0.000	0.000	0.000
15	0.149	0.200	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
16	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
17	0.086	0.200	0.000	0.000	0.261	0.000	0.000	0.000	0.000	0.000
18	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
19	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
20	0.000	0.000	0.000	0.032	0.000	0.000	0.000	0.000	0.000	0.000
21	0.000	0.202	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
22	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
23	0.000	0.000	0.000	0.000	0.088	0.000	0.000	0.000	0.000	0.000
24	0.000	0.000	0.000	0.000	0.059	0.000	0.000	0.000	0.000	0.000
25	0.000	0.000	0.000	0.000	0.588	0.000	0.000	0.000	0.000	0.000
26	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
27	0.000	0.000	0.000	0.000	0.569	0.000	0.000	0.000	0.000	0.000
28	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
29	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
30	0.000	0.000	0.000	0.073	0.000	0.000	0.000	0.000	0.000	0.000
31	0.000	0.073	0.206	0.000	0.000	0.000	0.000	0.000	0.000	0.000
32	0.000	0.000	0.000	0.000	0.289	0.000	0.000	0.000	0.000	0.000
…					…					
7205	0.000	0.000	0.000	0.000	0.354	0.000	0.000	0.000	0.000	0.000

	Topic									
Document	**31**	**32**	**33**	**34**	**35**	**36**	**37**	**38**	**39**	**40**
1	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001
2	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.289	0.000	0.000
3	0.000	0.000	0.000	0.000	0.000	0.000	0.342	0.000	0.099	0.000
4	0.000	0.000	0.000	0.000	0.000	0.000	0.286	0.000	0.000	0.000
5	0.000	0.000	0.000	0.000	0.000	0.380	0.000	0.000	0.000	0.000
6	0.000	0.000	0.000	0.000	0.000	0.281	0.000	0.000	0.000	0.000
7	0.000	0.000	0.000	0.000	0.000	0.449	0.000	0.000	0.000	0.000
8	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001
9	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001
10	0.000	0.000	0.222	0.000	0.207	0.000	0.000	0.053	0.000	0.000
11	0.000	0.033	0.478	0.096	0.000	0.000	0.000	0.000	0.000	0.000
12	0.000	0.000	0.000	0.111	0.000	0.000	0.057	0.000	0.000	0.000
13	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001	0.001
14	0.000	0.000	0.000	0.000	0.000	0.040	0.000	0.025	0.225	0.000
15	0.000	0.000	0.000	0.000	0.000	0.052	0.000	0.039	0.000	0.000
16	0.000	0.000	0.000	0.000	0.200	0.000	0.000	0.000	0.000	0.000
17	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
18	0.022	0.000	0.000	0.000	0.000	0.000	0.053	0.000	0.000	0.000
19	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
20	0.000	0.000	0.515	0.000	0.000	0.000	0.000	0.071	0.109	0.000
21	0.000	0.000	0.000	0.000	0.037	0.000	0.300	0.000	0.000	0.000
22	0.000	0.000	0.141	0.000	0.000	0.084	0.000	0.000	0.000	0.000
23	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.633	0.000	0.000
24	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.724	0.000	0.000
25	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
26	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
27	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
28	0.000	0.000	0.476	0.000	0.000	0.000	0.000	0.082	0.203	0.000
29	0.000	0.000	0.278	0.000	0.000	0.000	0.431	0.000	0.000	0.000
30	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
31	0.024	0.000	0.000	0.000	0.000	0.000	0.000	0.236	0.000	0.000
32	0.000	0.000	0.000	0.000	0.000	0.188	0.000	0.000	0.000	0.000
…						…				
7205	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.253	0.000	0.000

	Topic									
Document	**41**	**42**	**43**	**44**	**45**	**46**	**47**	**48**	**49**	**50**
1	0.001	0.001	0.001	0.001	0.001	0.971	0.001	0.001	0.001	0.001
2	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
3	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
4	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
5	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
6	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
7	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
8	0.277	0.224	0.001	0.001	0.001	0.001	0.475	0.001	0.001	0.001
9	0.349	0.229	0.001	0.001	0.001	0.001	0.398	0.001	0.001	0.001
10	0.000	0.000	0.000	0.124	0.000	0.000	0.000	0.000	0.000	0.000
11	0.000	0.000	0.000	0.089	0.000	0.000	0.000	0.000	0.000	0.000
12	0.000	0.000	0.000	0.127	0.261	0.000	0.000	0.000	0.000	0.000
13	0.001	0.721	0.001	0.001	0.001	0.001	0.001	0.001	0.179	0.001
14	0.000	0.000	0.000	0.126	0.000	0.000	0.082	0.000	0.000	0.000
15	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
16	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
17	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
18	0.000	0.000	0.000	0.342	0.000	0.000	0.496	0.000	0.000	0.000
19	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
20	0.000	0.000	0.000	0.061	0.000	0.000	0.000	0.043	0.000	0.000
21	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
22	0.000	0.000	0.000	0.086	0.000	0.000	0.000	0.000	0.000	0.000
23	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
24	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
25	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
26	0.000	0.000	0.000	0.180	0.000	0.000	0.000	0.000	0.000	0.000
27	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
28	0.000	0.000	0.000	0.147	0.000	0.000	0.000	0.000	0.000	0.000
29	0.000	0.000	0.000	0.278	0.000	0.000	0.000	0.000	0.000	0.000
30	0.000	0.000	0.472	0.000	0.000	0.000	0.000	0.000	0.000	0.000
31	0.000	0.000	0.000	0.327	0.000	0.000	0.000	0.000	0.030	0.000
32	0.000	0.000	0.000	0.196	0.000	0.000	0.000	0.000	0.000	0.000
…						…				
7205	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

APPENDIX D

SME REVIEW OF FLAGGED ITEM PAIRS

Rater	Items Reviewed	Enemies	% Enemies
SME 1	148	38	25.7%
SME 2	230	66	28.7%
SME 3	340	151	44.4%
SME 4	496	137	27.6%
SME 5	137	37	27.0%
SME 6	297	31	10.4%
SME 7	194	68	35.1%
SME 8	1065	147	13.8%
SME 9	405	122	30.1%
SME 10	365	107	29.3%
SME 11	449	104	23.2%
SME 12	147	56	38.1%
SME 13	151	101	66.9%
SME 14	344	65	18.9%
SME 15	361	62	17.2%
SME 16	195	42	21.5%
SME 17	139	80	57.6%
SME 18	110	14	12.7%
SME 19	322	110	34.2%
SME 20	85	13	15.3%
SME 21	2373	325	13.7%
SME 22	276	36	13.0%
SME 23	131	22	16.8%
SME 24	188	9	4.8%
SME 25	60	25	41.7%
SME 26	94	13	13.8%
SME 27	167	30	18.0%
SME 28	23	8	34.8%
SME 29	17	7	41.2%
Total	**9309**	**2026**	